HCLSoftware

HCL MyXalytics

FinOps Admin Guide

Version 6.6

HCL has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the HCL website at www.hcltechsw.com.

Table of Contents

Table of Figures

List of Tables

Document Revision History

This guide is updated with each release of the product or when necessary.

This table provides the revision history of this User Guide.

Version Date	Description
January, 2025	HCL_MyXalytics_FinOps Admin Guide_v6.5
May, 2025	HCL_MyXalytics_FinOps Admin Guide_v6.6

1 Preface

This section provides information about the MyXalytics FinOps Admin Guide and includes the following topics:

- Intended Audience
- About this Guide
- Related Documents

1.1 Intended Audience

The document is intended for users involved with data reporting such as data reporters.

1.2 About this Guide

This guide contains an overview of FinOps. It summarizes MyXalytics dashboard and FinOps admin modules.

1.3 Related Documents

The following documents can be referenced in addition to this guide for further information on MyXalytics.

- Introduction to MyXalytics Guide
- MyXalytics User Guide
- MyXalytics Troubleshooting Guide

1.4 Conventions

The following typographic conventions are used in this document:

Table 1 – Conventions

Convention	Element
Boldface	Indicates graphical user interface elements associated with an action, or terms defined in text or the glossary
Underlined blue	Indicates cross-reference and links
Italic	Indicates document titles, occasional emphasis, or glossary terms
Courier New (Font)	Indicates commands within a paragraph, URLs, code in examples, and paths including onscreen text and text input from users
Numbered lists	Indicates steps in a procedure to be followed in a sequence
Bulleted lists	Indicates a list of items that is not necessarily meant to be followed in a sequence

2 FinOps Overview

In a rapidly digitizing business paradigm, cloud platforms have emerged as an essential element for future-facing enterprises. This is also seeing enterprises move beyond private and public cloud platforms and opt for a hybrid cloud model for their business. Hybrid cloud offers you a range of benefits, including improved choice of services, increased speed of implementation and execution, and greater scalability. However, it also poses challenges around cloud cost visibility, optimization, and governance. Poor financial management, inadequate reporting capabilities and inability to predict the OPEX costs accurately are the realities of today's multi-cloud environments.

MyXalytics – FinOps our AI-driven Cloud FinOps Visibility and Insights product— helps you effectively visualize, manage, and optimize your multi-cloud spends.

Advantages of FinOps-

1. **Unified Cloud Cost Visibility:** With granular visibility into all your cloud environments, capture, track the usage patterns over time, and leverage AI and Machine Learning to accurately forecast budgets and minimize deviations.
2. **Cost Anomaly Detection:** Leverage AI-driven outlier analysis to identify anomalous cloud costs and get alerted via automated notifications using various channels like ITSM, Slack, ChatOps and others.
3. **Cost Optimization:** Reduce the amount of time spent on analyzing and optimizing your cloud spends by leveraging various techniques like rightsizing, scheduled instances, reserved / committed usage instances, waste elimination, and others.
4. **Cloud Governance:** Ensure continuous security, compliance, and governance across your cloud environments through native advisory and security visibility and insights.
5. **Task Allocation and Tracking:** Analyze and Identify problem areas, and assign them to concerned teams for resolution, with continuous tracking and updates.
6. **Kubernetes Inventory and Overview:** Comprehensive view of cluster, node, namespace, and pod inventory counts, seamlessly integrated with their associated billing details and performance metrics.
7. **Serverless Function Overview:** Detailed view of serverless functions, showcasing their inventory counts alongside related billing information and performance metrics.

3 FinOps Management

3.1 Budget Creation

This module helps in configuring the Budget of a customer based on selected cloud provider.

Budget can be defined on three levels:

- Overall
- Subscription
- Service
- Application
- BusinessUnit
- Environment
- Region

Steps to configure the budget:

1. Go to Configuration Tab - > FinOps -> Budgets.

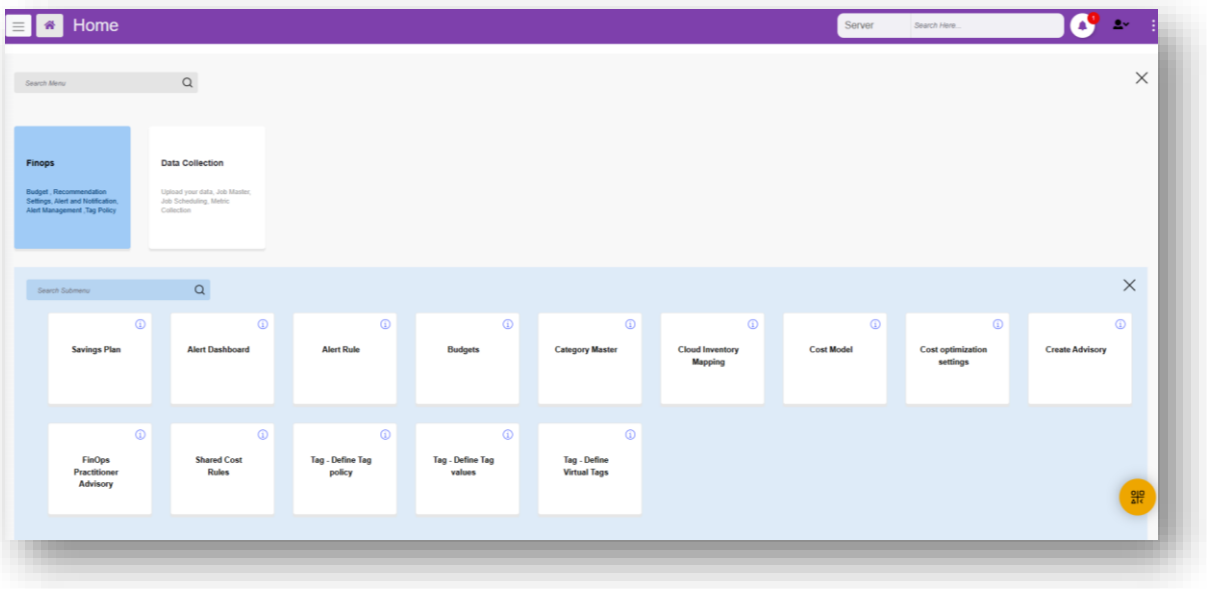

Figure 1 – Budget Configuration

2. Click on ‘Create Budget’ button on top right corner.

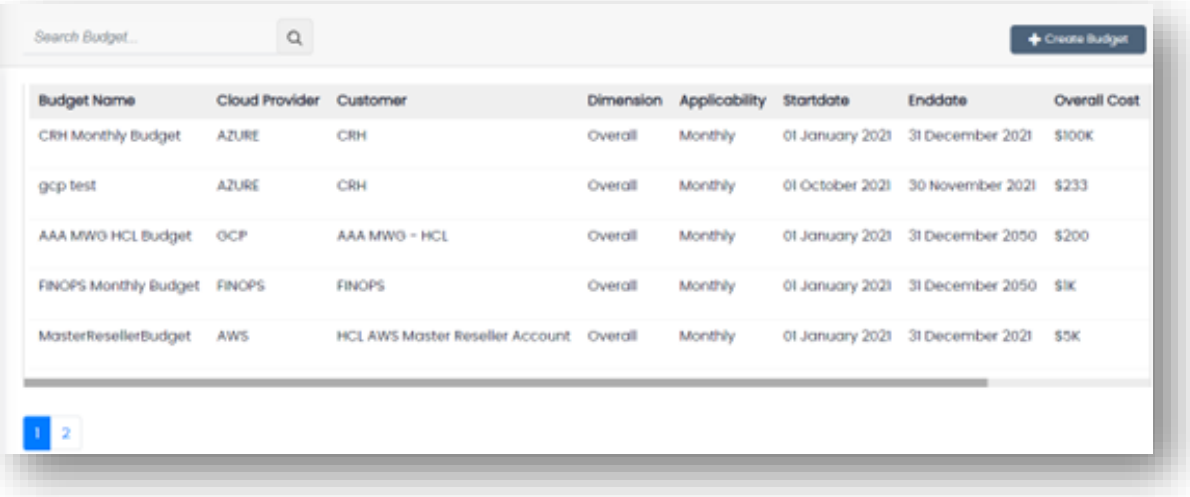

Figure 2 – Create Budget

3. In the below page, following fields should be filled in:

- **Budget Name**: Enter a name for the Budget. The budget list will contain all the budgets created for the user's engagement logged in.
- **Provider**: Select cloud provider i.e. : AZURE, AWS & GCP.
- **Currency**: selected currency will be used for creating budget.
- **Dimensions**: Budgets can be defined in several dimensions like: Overall (Engagement level), Subscriptions, Services etc. Based on selected dimensions, budgets can be allocated.

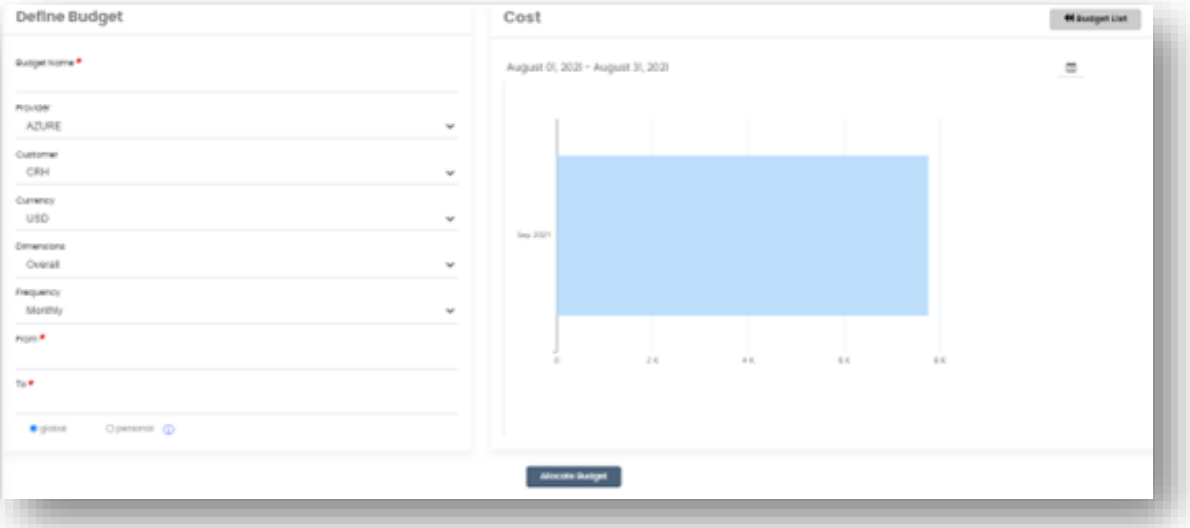

Figure 3 – Define Budget

– When '**Overall'** option is selected in dimensions, it shows the estimated cost of the current month and Average Usage (3 months) to determine the value of budget. This further helps in creating the budget.

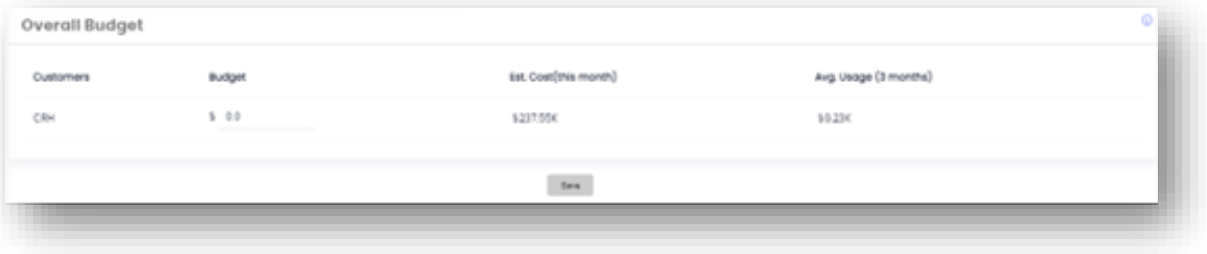

Figure 4 – Overall Budget

Also shows the current monthly cost trend of the current month.

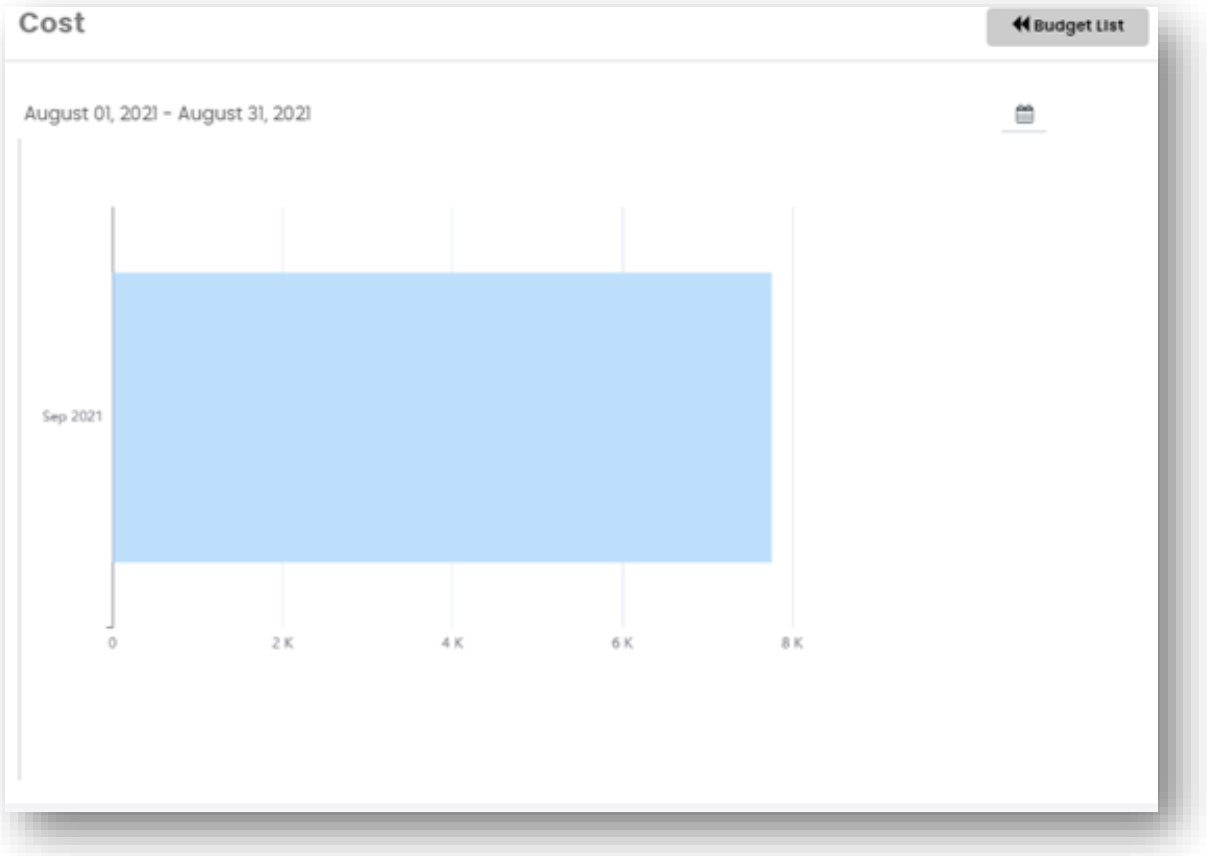

Figure 5 – Cost (Budget Configuration)

- When Subscription level is selected, all subscription related to the customer are shown.

Figure 6 – Overall Budget

- Also shows the cost trend subscription wise for the current month.

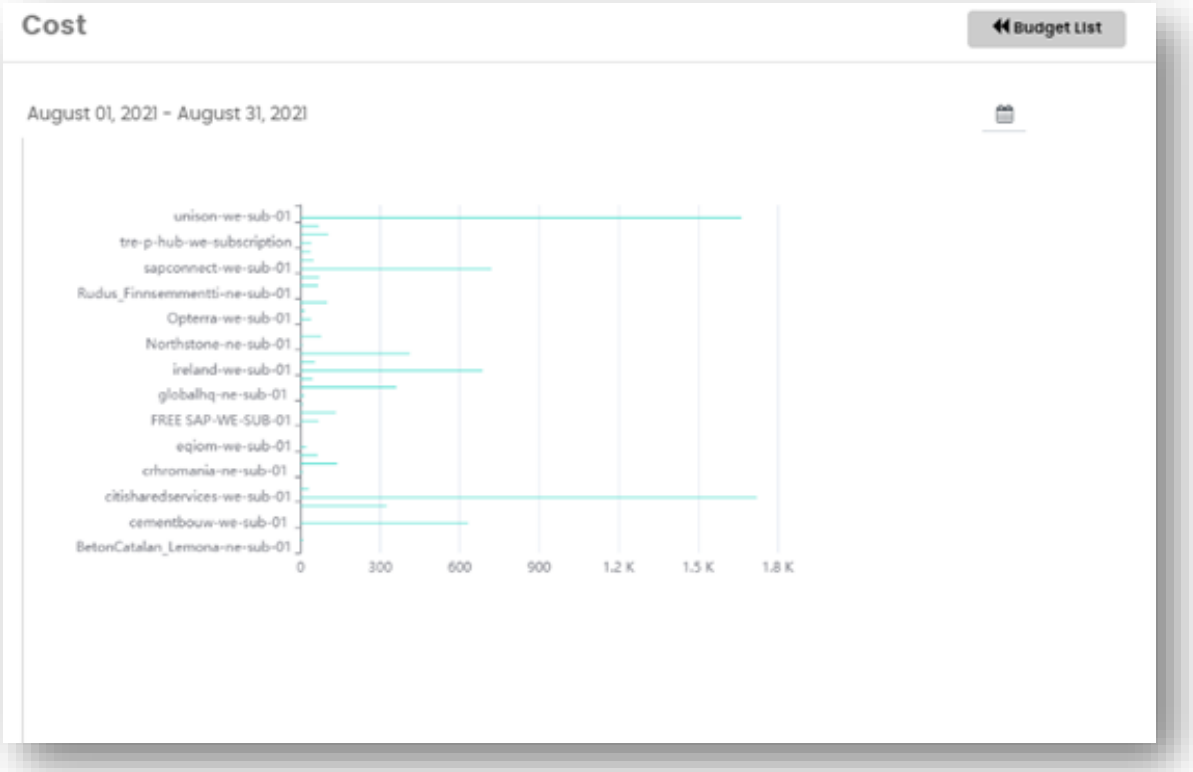

Figure 7 – Cost Graph (Budget Configuration)

- When selected service wise dimension for creating budget:

Figure 8 – Budget Configuration (Cont.)

– It also shows the subscription-wise cost trend for the current month.

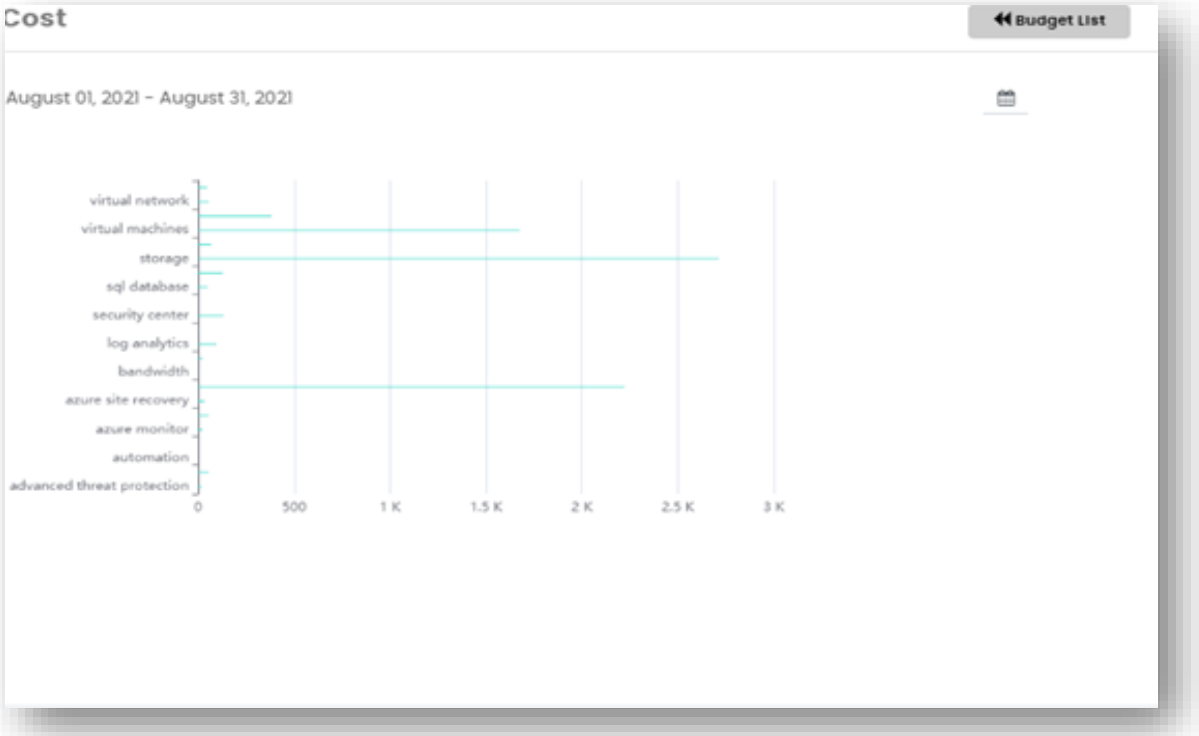

Figure 9 – Cost Table (Budget Configuration)

- Validity of budget: Selection of from and to date
- Global/Personal: Global rules can be configured when user needs to create budget visible to all inside a particular engagement, whereas Personal rules can be configured at User level.

4. Click on the Save button to save. Once saved, the created budget is now visible for edit and delete.

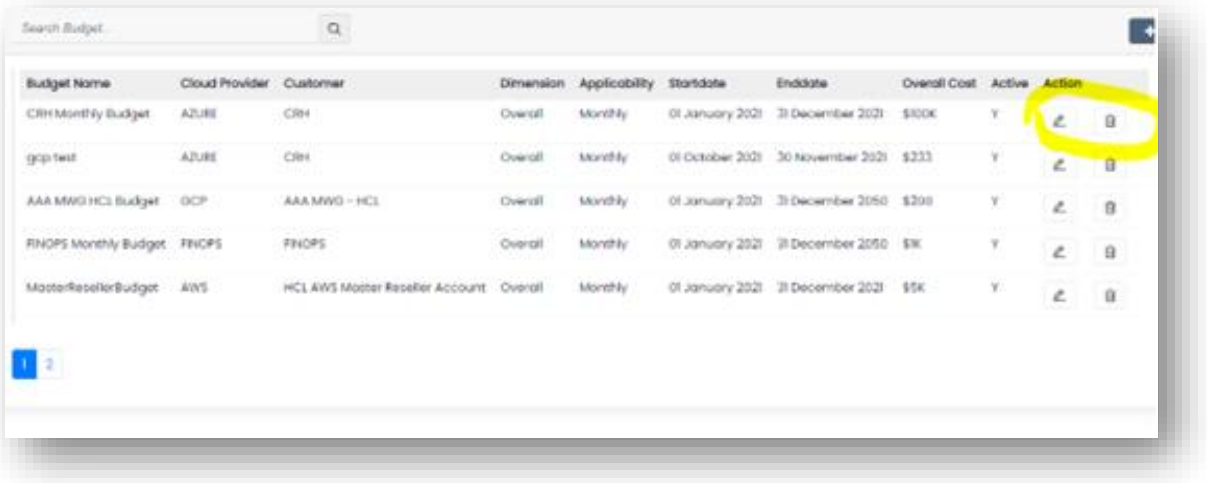

Figure 10 – Search Budget

4 Tag Policy

Steps to configure tag policy:

1. From the left menu, go to Configuration -> FinOps -> Tag Policy

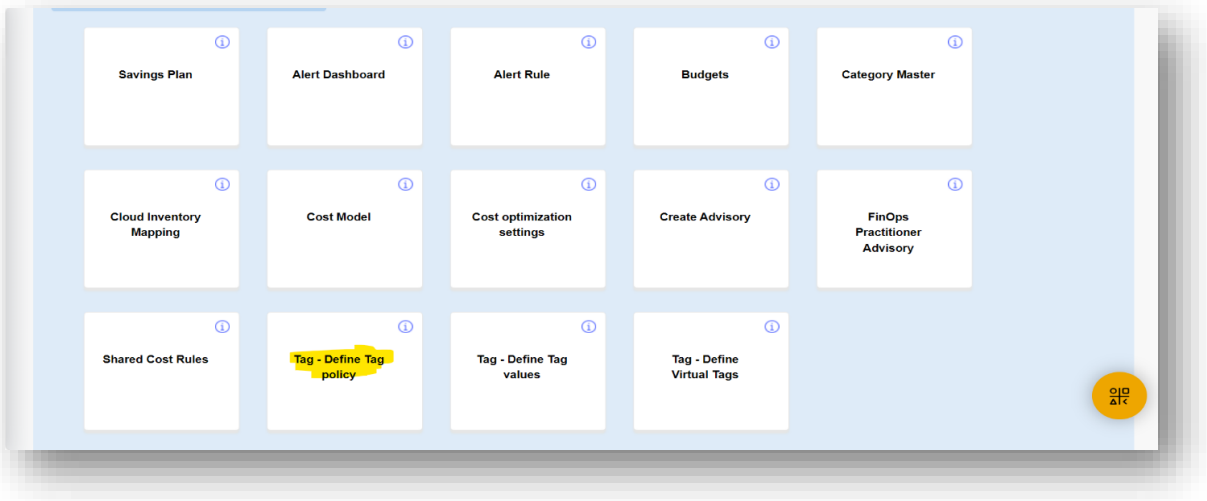

Figure 11 – Tag – Define Tag Policy

2. Click on Add tag policy button and configure the basic details of Policy like cloud provider, Customer, policy name, level (Global, Service, Resource Group) on the below screen-

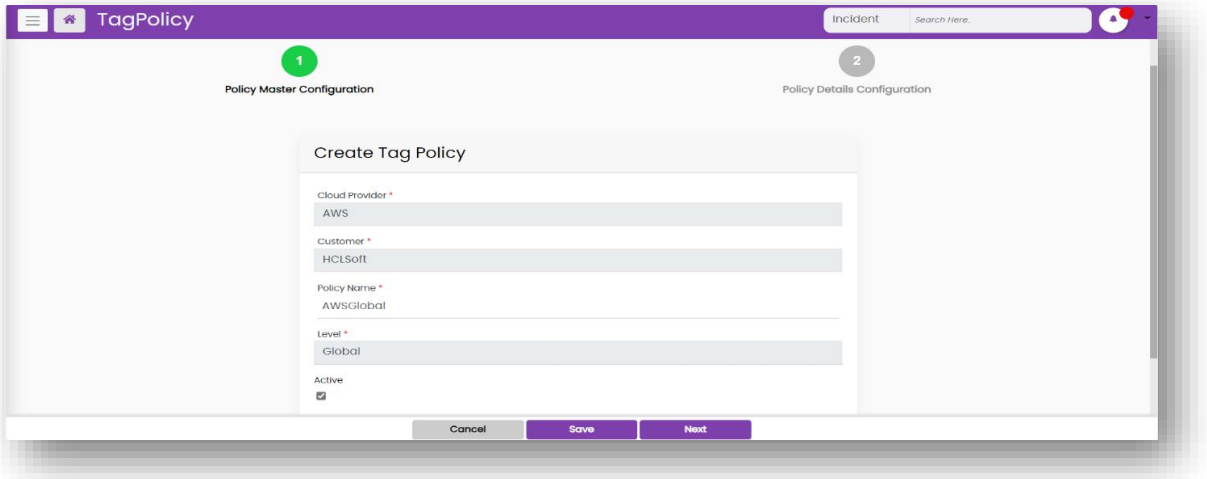

Figure 12 – Tag Policy (Cont.)

3. Click on save and next. Configure the tag and level (Service Resource Group) details in the below-screen-

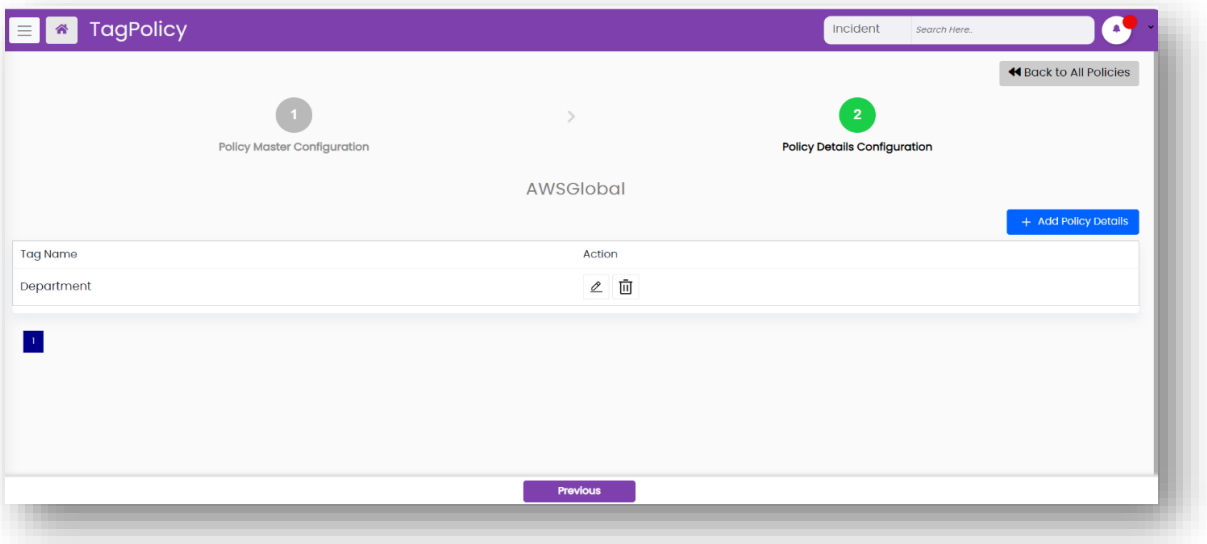

Figure 13 – Tag Policy (Cont.)

4. Multiple tags can be configured for which list is displayed on the same screen.
5. Schedule the job compliance collection from job scheduling screen and configure the reports.

4.1 Virtual Tag

In this console's initial page, you'll find a comprehensive overview of cloud providers, their respective services, and subscriptions, resource count, Global Tag Noncompliance percentage, Service Tag Noncompliance percentage nested within those services. In the second page, you will be able to map the tag values with resources for both global and service-level tags. Additionally, you can observe the percentage of Tag noncompliance.

Steps to access console and stored procedures used:

1. To access the Virtual Tag page, Configuration -> FinOps -> Tag – Define Virtual Tags

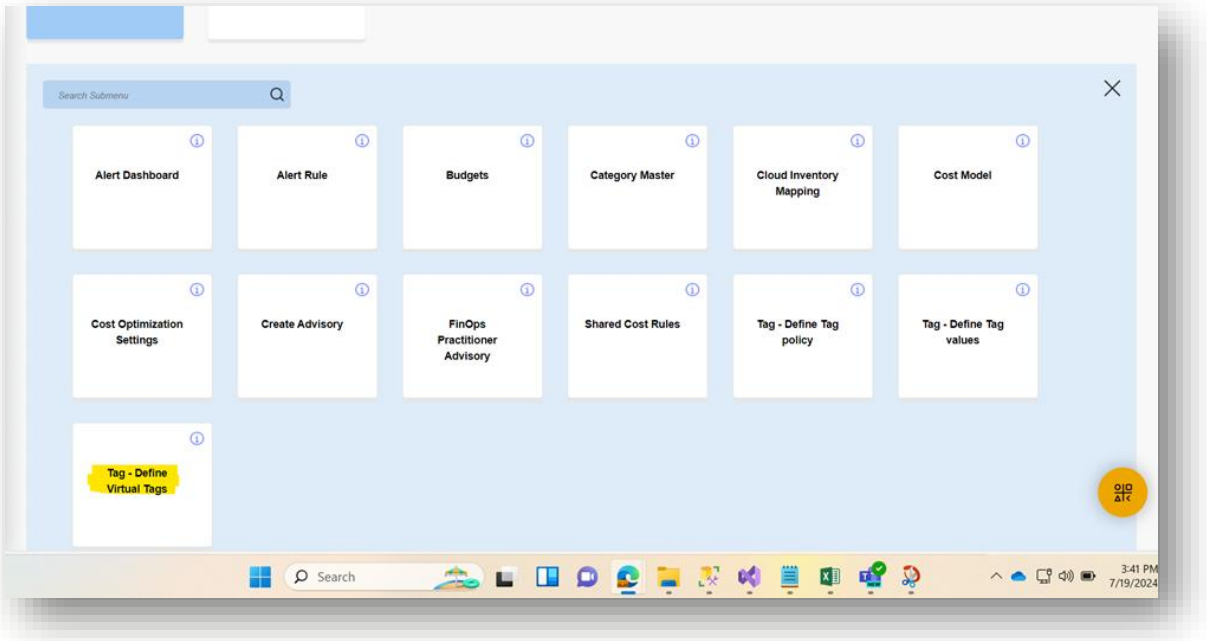

Figure 14 – Tag – Define Virtual Tags

2. This is the landing page, where you can see the cloud providers at the top. These cloud providers are populated by default. The first cloud provider will be selected and the services, subscriptions under each service, resource count, Global Tag Noncompliance percentage and service Tag Noncompliance percentage will be shown. The services, subscriptions, resource count and Noncompliance percentage data are getting and click on action add button beside subscription to map the resource's tags with tag values.

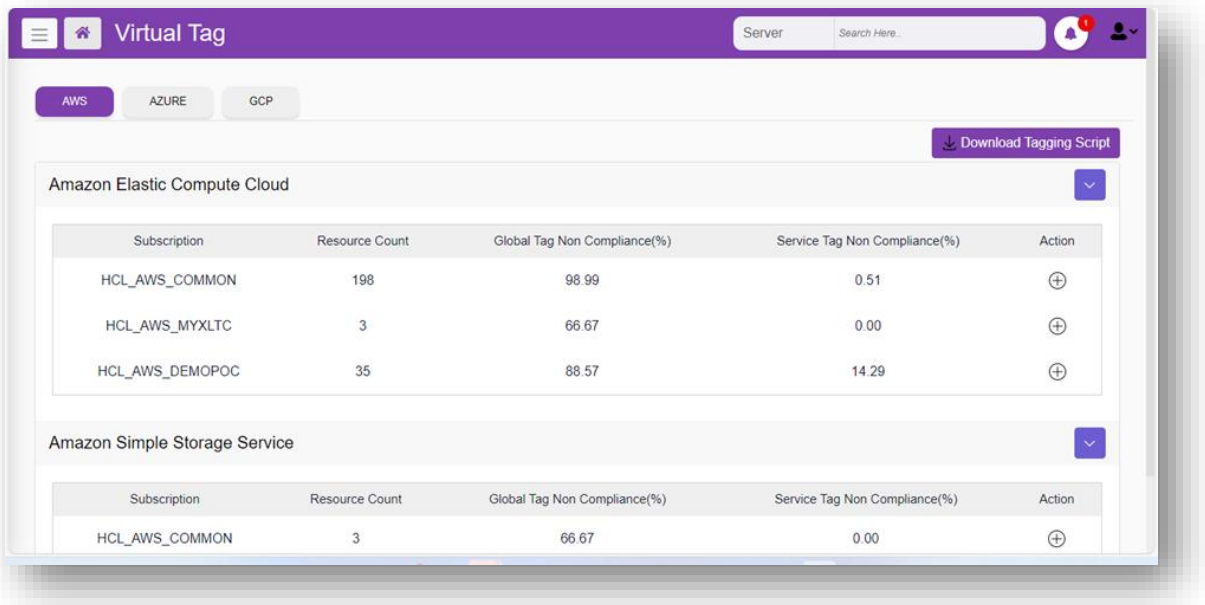

Figure 15 – Define Virtual Tags (Cont.)

3. In the top right corner, you will have the button Download Tagging script to download the entire Tag Script.
4. For the resources there are already cells populated with tag values that are compliant. And the cells which are showing red color then the corresponding resources are non-compliant.

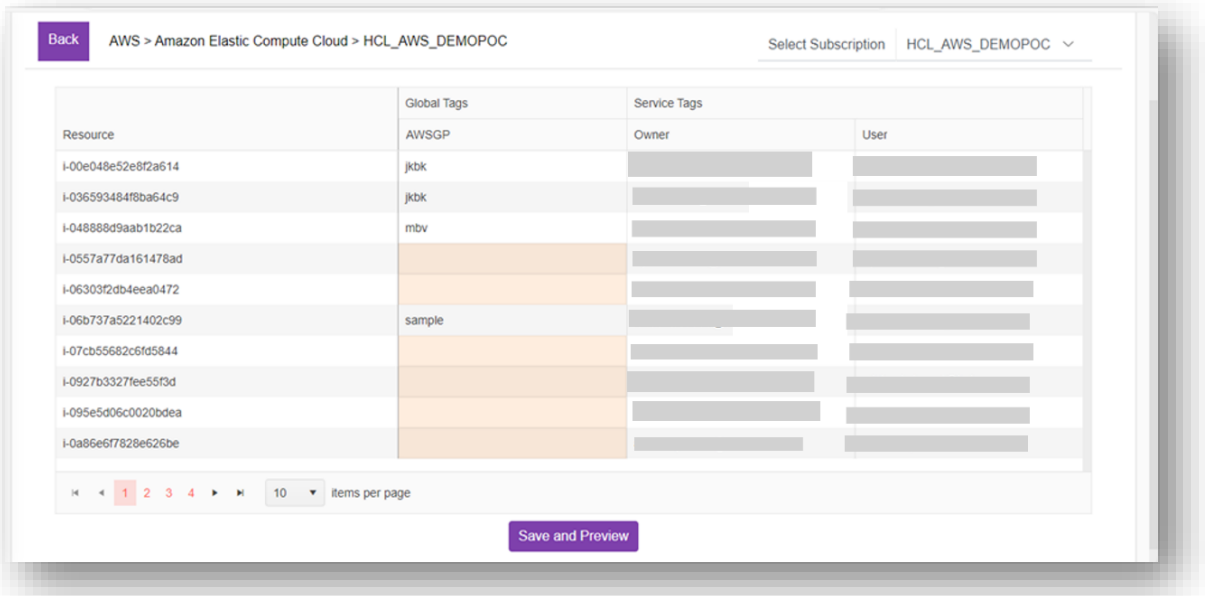

Figure 16 – Define Virtual Tags (Cont.)

5. In the above figure, the grid used is kendo and the resources list is fetched.

6. In the top right there is subscription dropdown, you will be able to change the subscription under the service. Clicking on each cell under the tags will show the dropdown with the tag values that are added in addtagvalue console. Select the tag values and click on save.

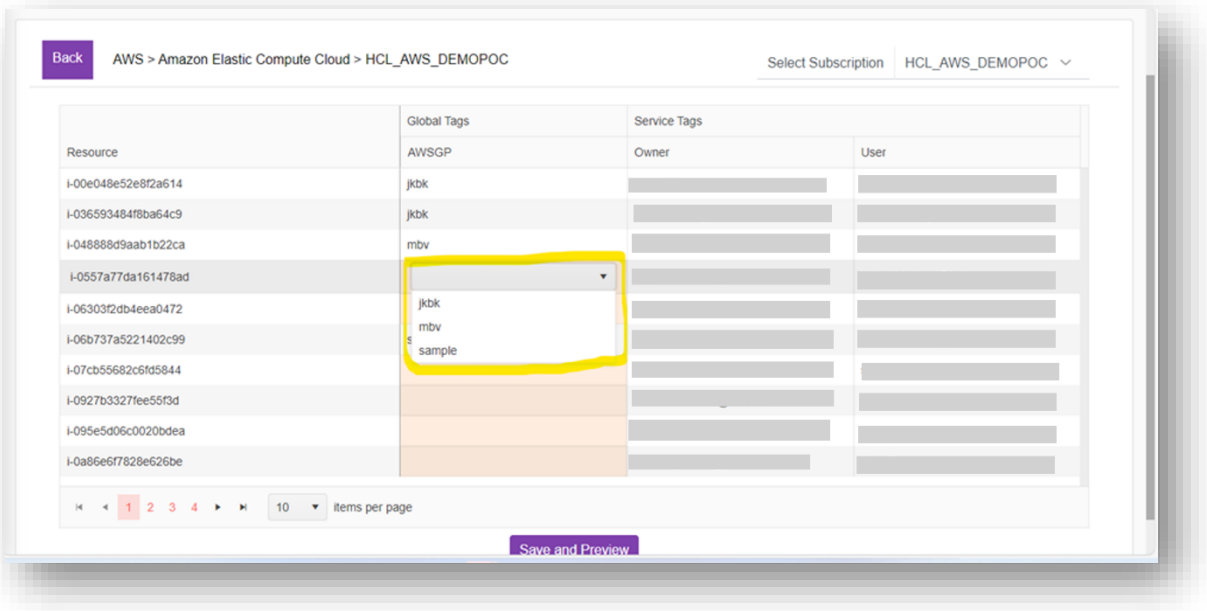

Figure 17 – Define Virtual Tags (Cont.)

Notes:

Before accessing the Virtual tag console, please make sure that for the corresponding engagement there are already tag policies (Global & Service Level) created with tag names in tag policy configuration page.

Also, please make sure that you have added tag values in the AddTagValues console for the Tag names that are created in the tag policy configuration console.

AddTagValues or TagPolicy buttons are available on top right to navigate to these consoles.

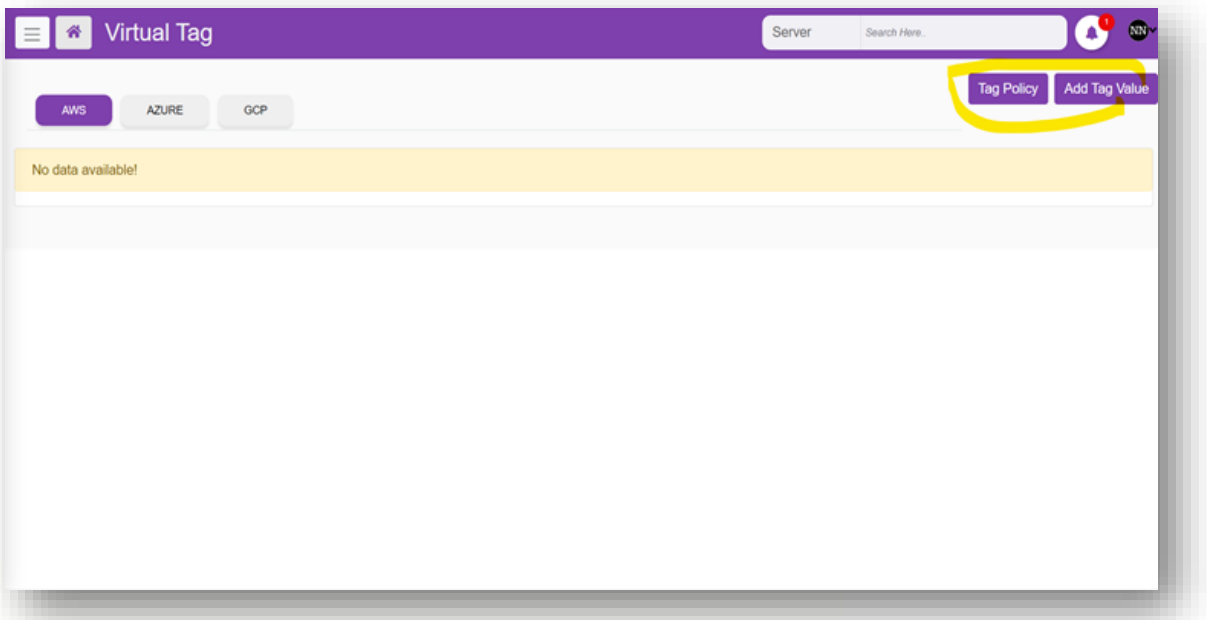

Figure 18 – Virtual Tags

4.1.1 Add Tag Values Console

In this console, you can assign values to the global and service tags for cloud provider that are already configured in the tag policy configuration page.

1. To access the Add Tag Values page navigate to menu - > Configuration- > FinOps - > Tag - Define Tag values

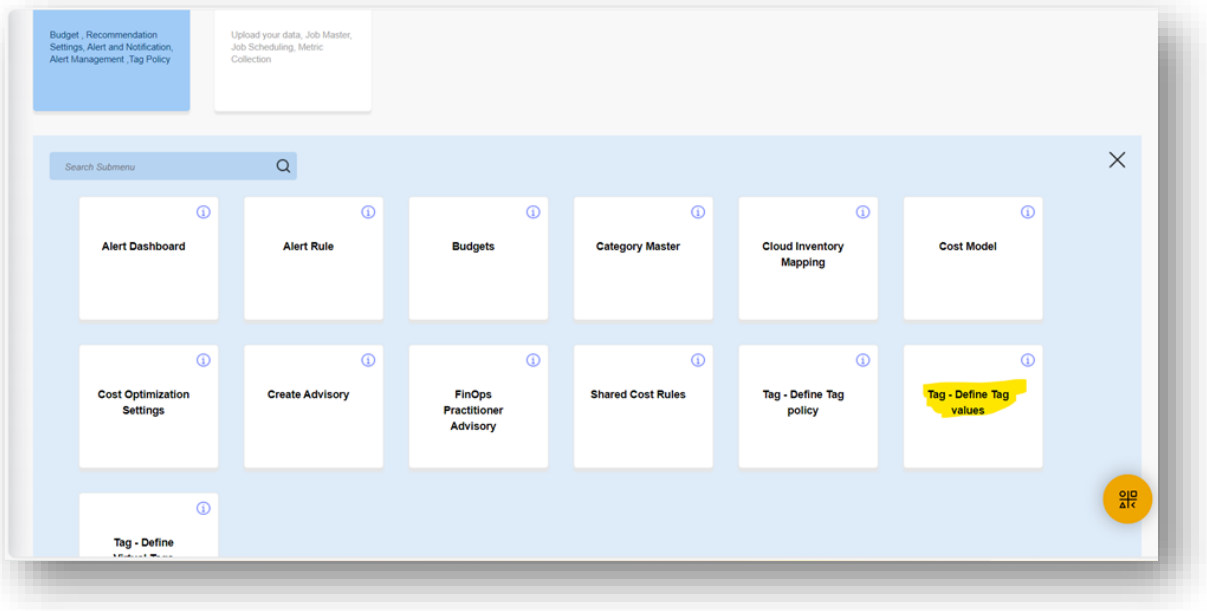

Figure 19 – Define Tag Values

2. This is the landing page of AddTagValues Console. Here, you could see all the tags that are created in the Tag policy configuration page. Click on add button to add the Tag values for the corresponding tag name.

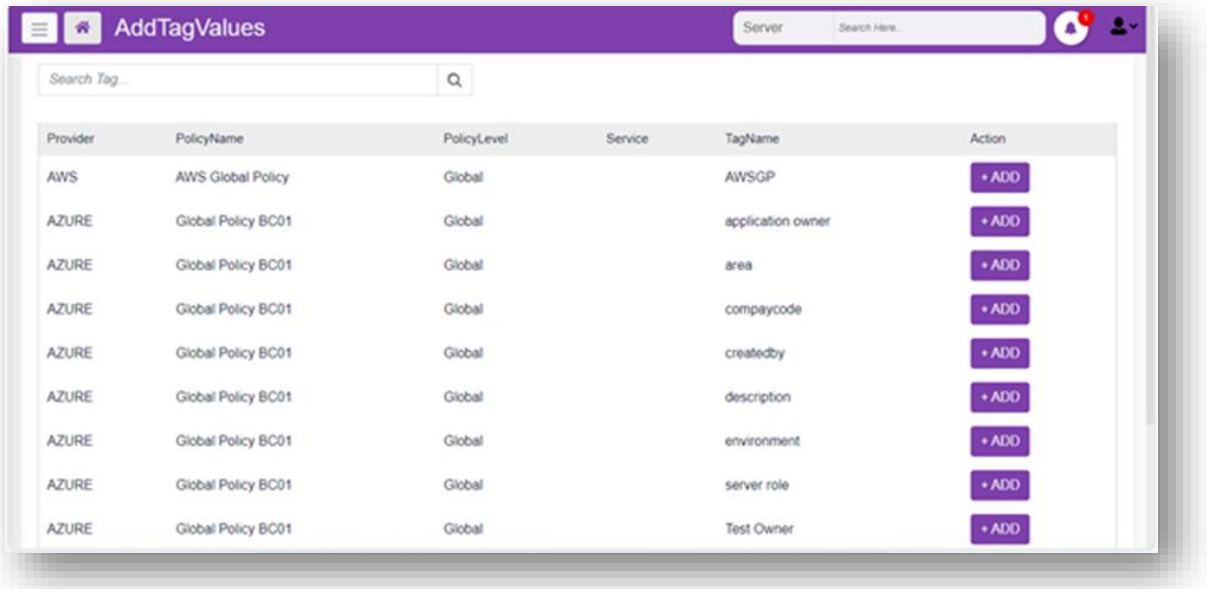

Figure 20 – Add Tag Values

3. In the below page you can see already existing tag values for the tags which are fetched, and the last row will have empty text box to enter new tag value. You can update, delete the existing tag values and add new tag values. For saving, editing, and deleting the tag value.

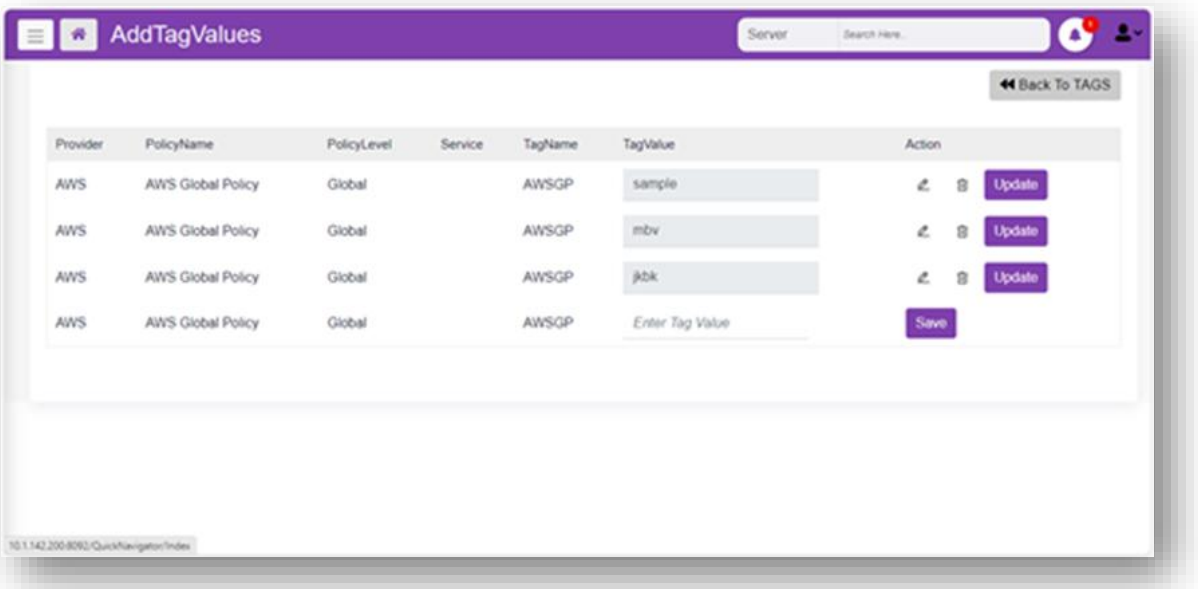

Figure 21 – Add Tag Values (Cont.)

4. Added Two buttons, for navigating Virtual Tag or TagPolicy Pages.

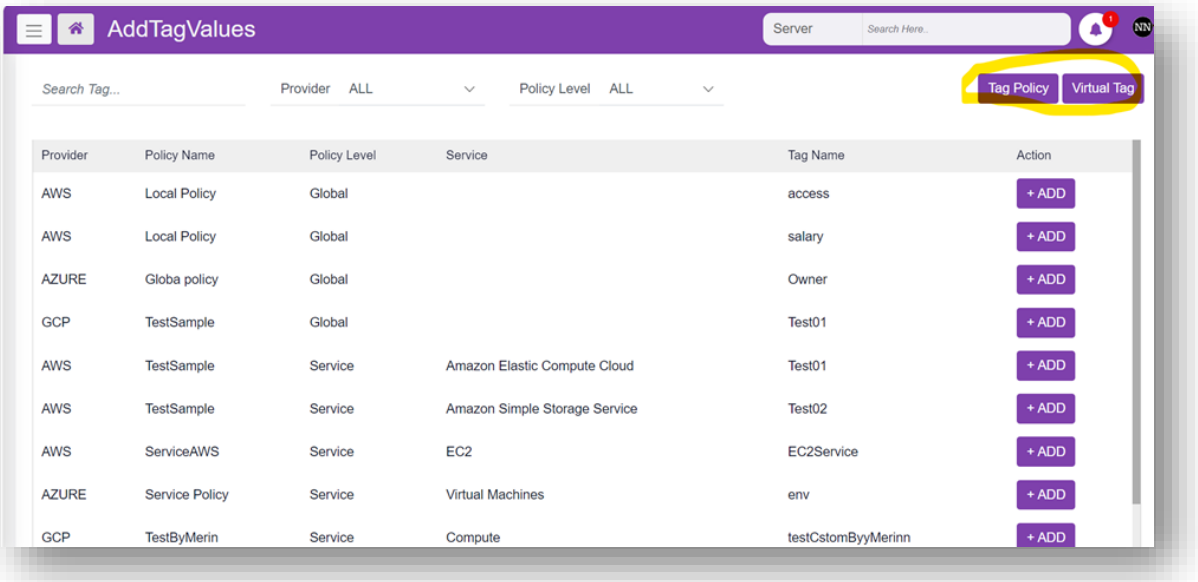

Figure 22 – Add Tag Values (Cont.)

4.2 Cost Optimization Settings

This module helps user to optimize and save their cost for different infrastructural demand. It works on different parameters like Cloud Provider, Offered Service, Service Category, Optimization Technique Category, Technique, Technique Rule etc. Users can choose within the list of available cloud providers and optimize the cost of its infra. Please find below the detailed explanation with images-

1. To access the Cost Optimization Settings Page, navigate to menu -> Configuration -> FinOps -> Cost Optimization Settings

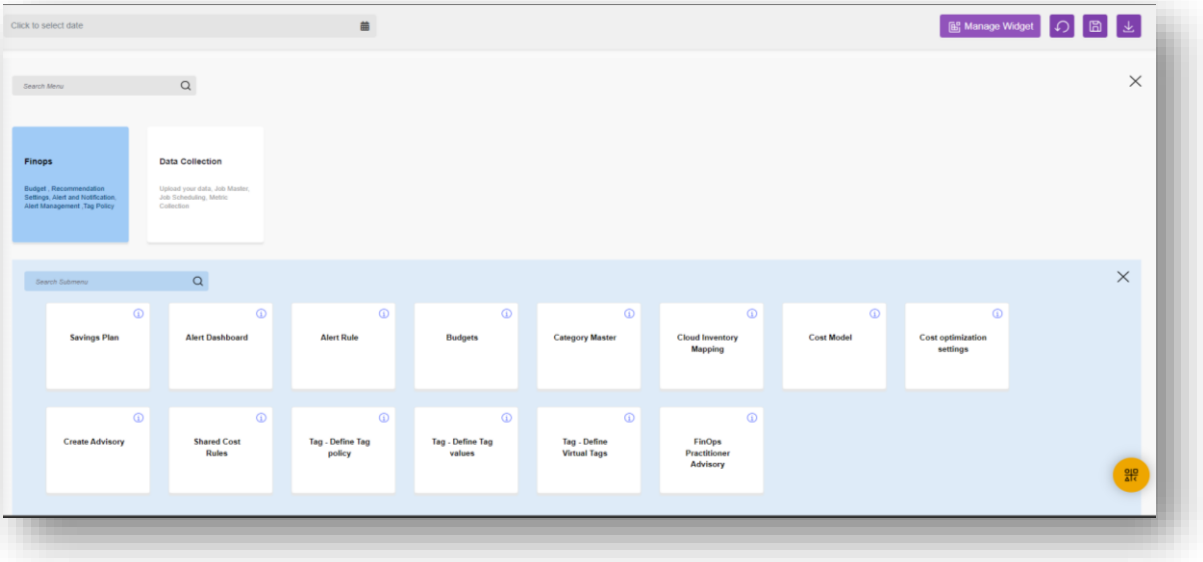

Figure 23 – Cost Optimization Settings

2. Clicking on the Cost Optimization Settings option, the user will be navigated to Cost Optimization Settings page.

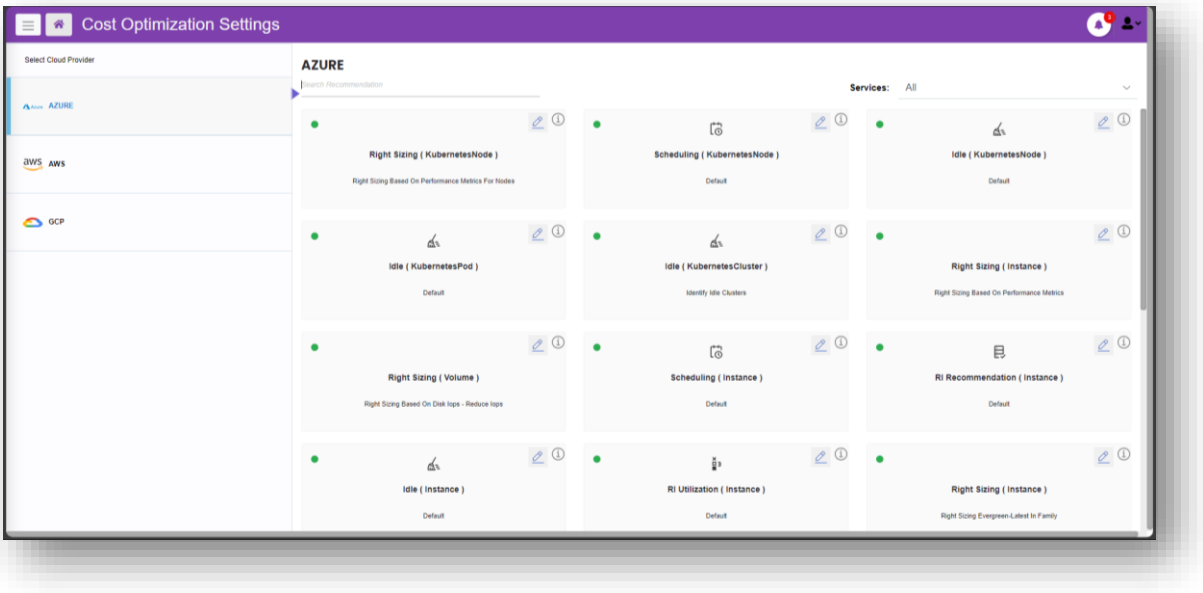

Figure 24 – Cost Optimization Settings (Cont.)

The above page has two sections. One section contains the list of available cloud providers, and another section contains the list of available Technique and Technique Rule within that cloud provider. Users can search with the name of the technique by using the search option available and categories the Technique with respect to the Instance or the Volume. Each technique has its own set of rules and options. To edit the given Technique user can click on the edit. Please find below image for Right Sizing technique after clicking on Edit button :

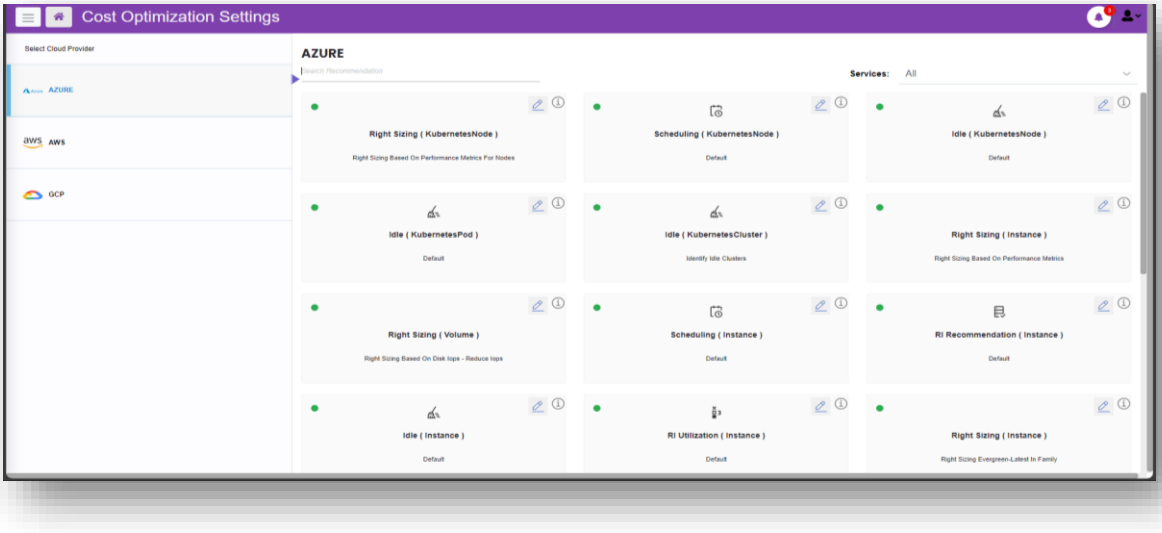

Figure 25 – Cost Optimization Settings (Cont.)

Screen 1: It is used to configure the different KPI's. KPI Configuration completely depends upon Technique and Technique rules for each service (types of resource). In the below image, we can see different parameters like Within Service Type, Within Family Type, Within Region etc. These parameters are predefined according to the Technique and Technique rules. User can Save its recommendation by clicking the Save button at the bottom of the page shown in the below image.

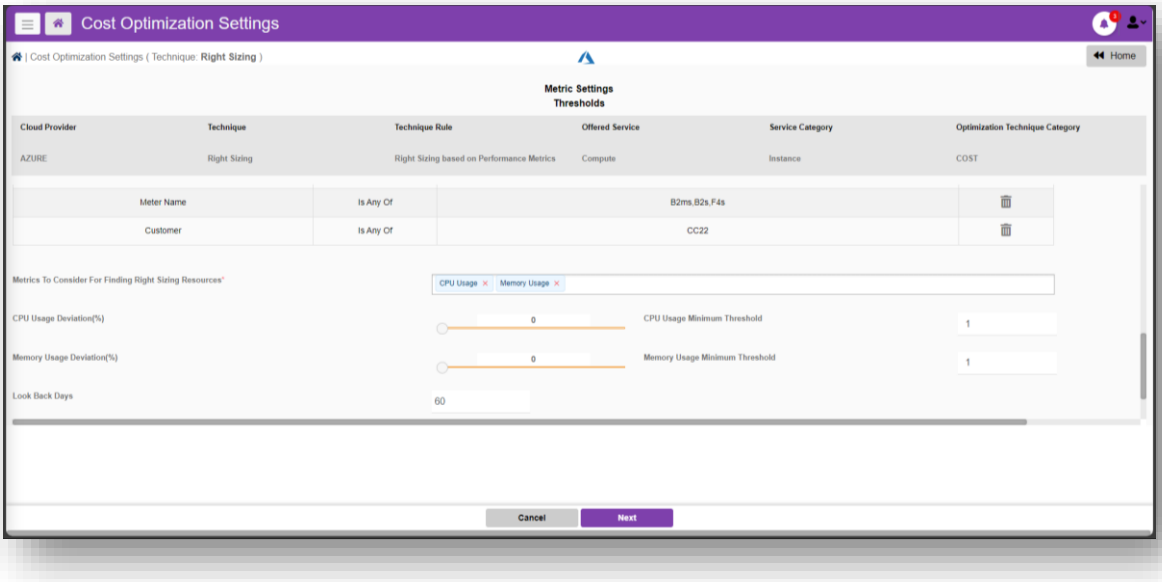

Figure 26 – Cost Optimization Settings (Cont.)

Screen 2: By clicking on the Next button, the user will navigate to this section. This section is used if users want to perform some advance actions on KPI's like Average, Maximum, Minimum on the specific parameter and can save them.

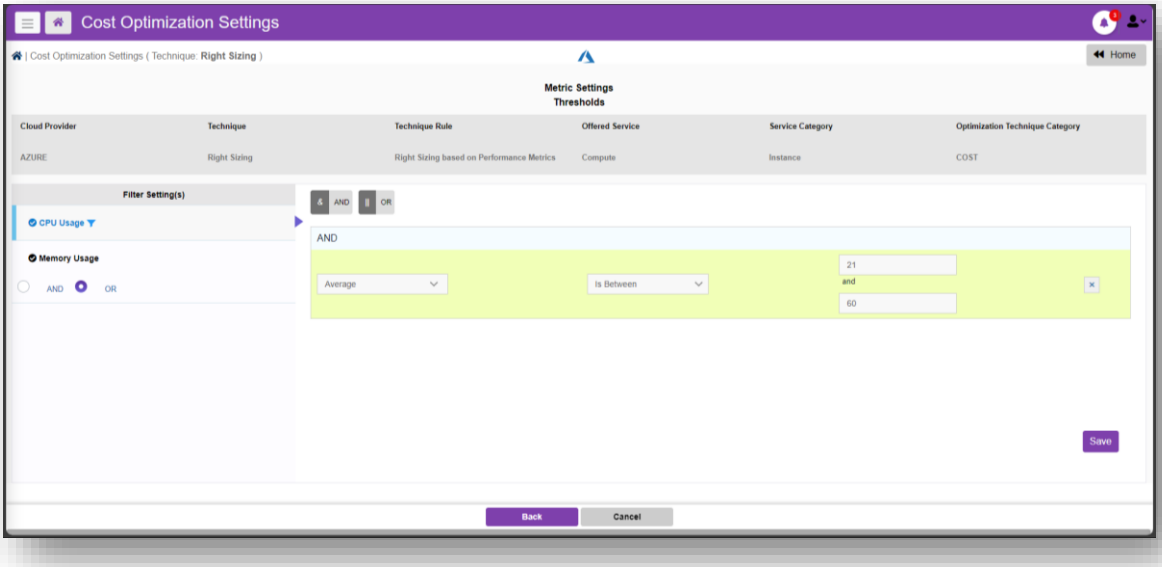

Figure 27 – Cost Optimization Settings (Cont.)

AWS – Exclusion filter for DBinstance and Instance recommendation

- For AWS recommendation we have introduced a new filter which will enable the user to exclude Nano machines or any other parameters, i.e., Nano, macro... from the front end in the cost optimization settings page.
- We have also enabled the filters like Instance Category, Instance Family and Family size params to the recommendations for the AWS RDS DBInstance which will again be used as a filter in the recommendation where users can include or exclude any of the given categories.

Select the AWS instance for the rightsizing based on the performance metric:

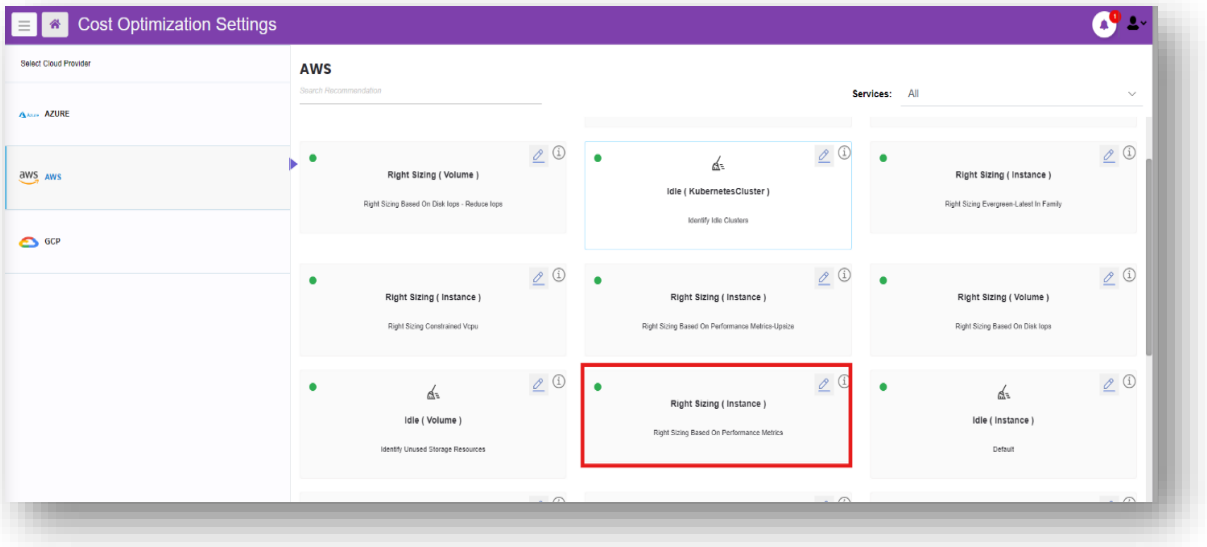

Figure 28 – Cost Optimization Settings (Cont.)

Select the filter where we can choose the Family Size for any required param Ex:-NANO:

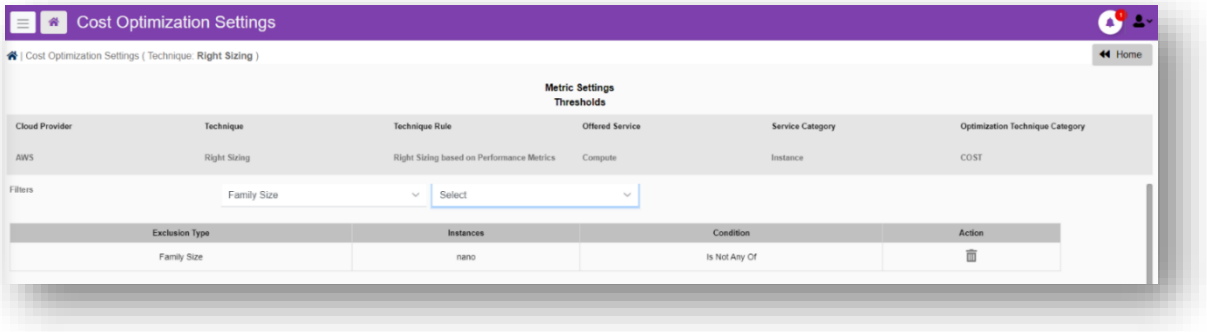

Figure 29 – Cost Optimization Settings (Cont.)

For the DBInstance filter settings, select the DBINSTANCE rightsizing based on the Performance Metrics:

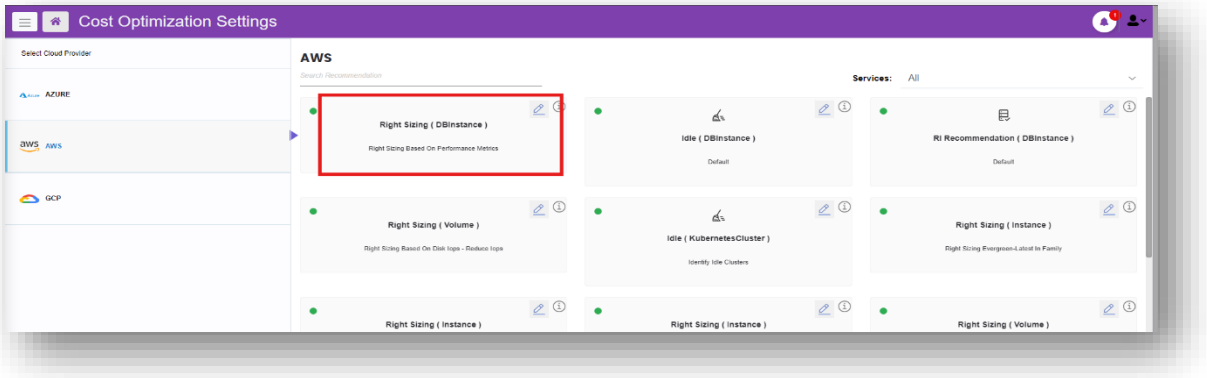

Figure 30 – Cost Optimization Settings (Cont.)

We can select any required params like Instance Category, Instance Family and Family Size acts as a filter for the recommendation:

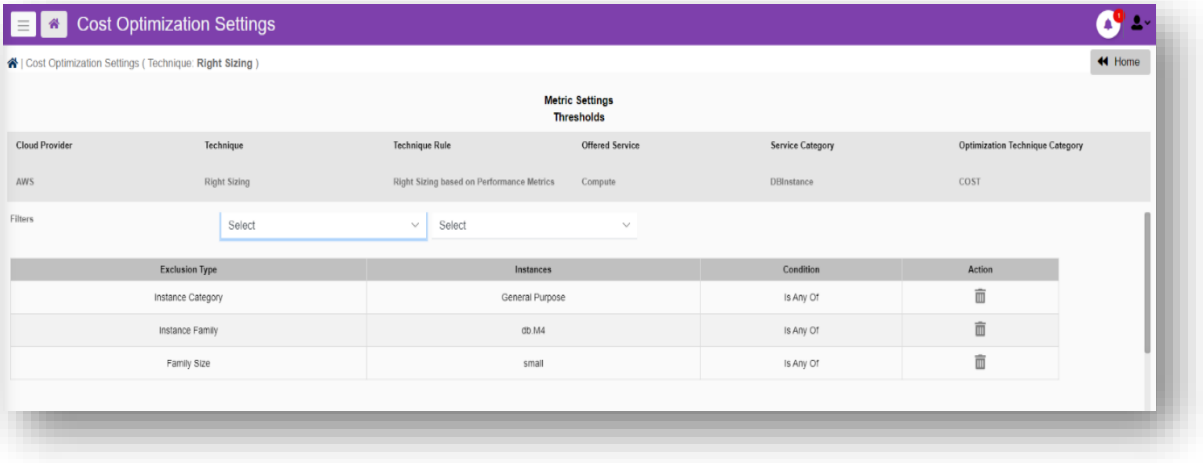

Figure 31 – Cost Optimization Settings (Cont.)

4.3 Alerting

Users can go to alert dashboard by following below path:

1. Menu -> Configuration -> FinOps -> Alert Dashboard

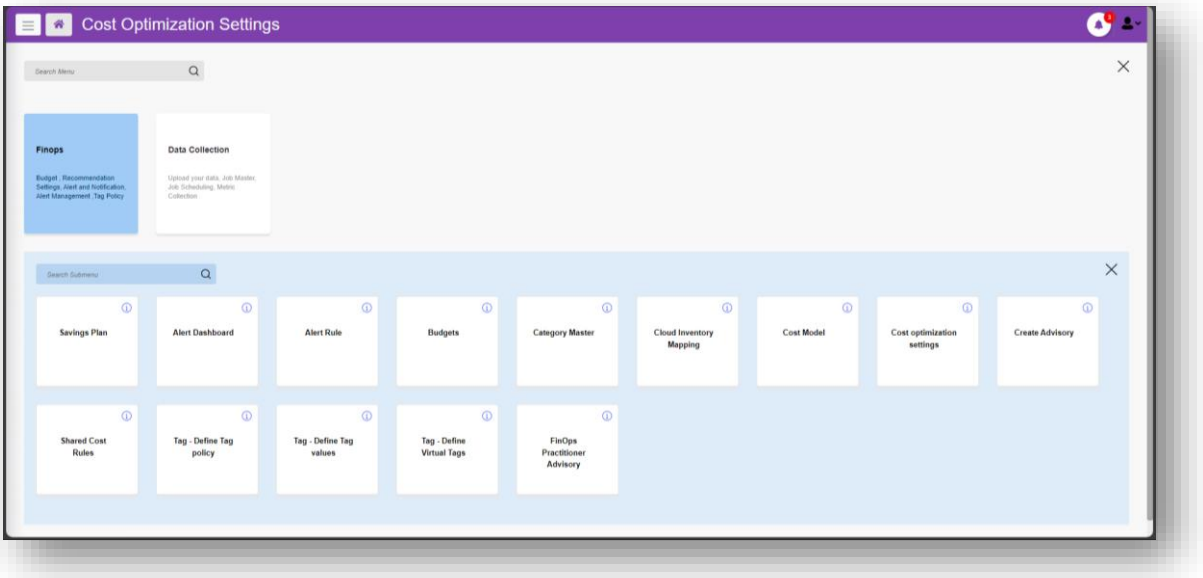

Figure 32 – Alert Dashboard

2. The Alert Dashboard lists the alerts created for -
   a. Recommendations from Cost Optimization view
   b. Alert rule configured for Cost Anomaly
   c. Alert rule configured for Cost
   d. Alert rule configured for Budget
3. As soon as user clicks on Alert Dashboard, user can see the below screen:

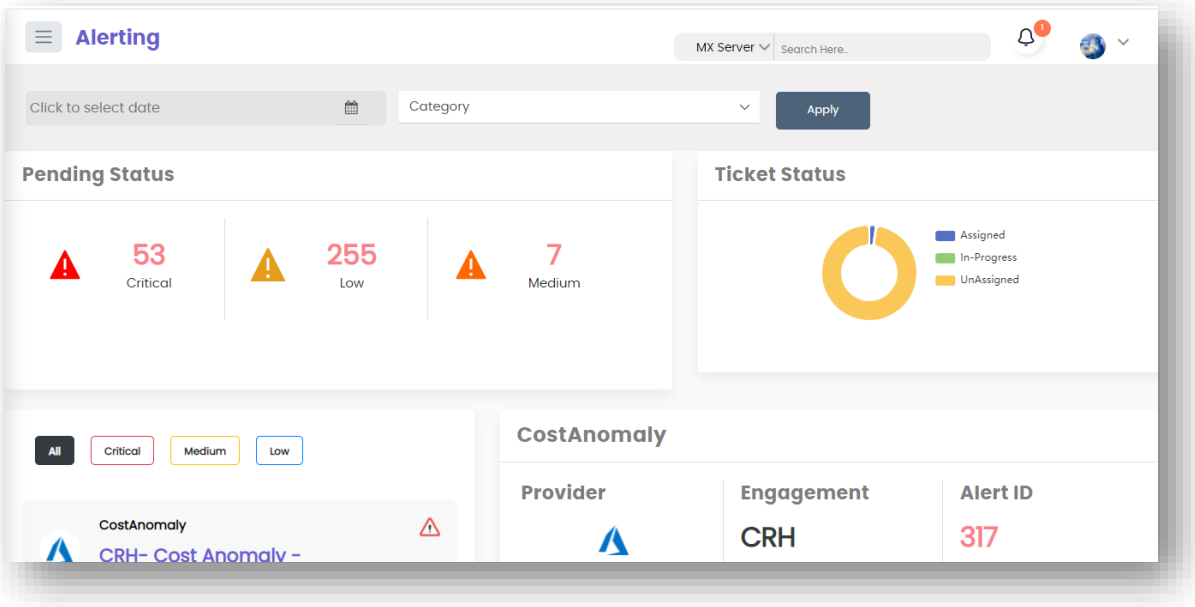

Figure 33 – Alert Dashboard (Cont.)

4. In this dashboard, users can have a look at alerts which are in:

– Pending status severity wise

– Count of alerts status wise.

5. Users can filter alerts based on creation date and category:

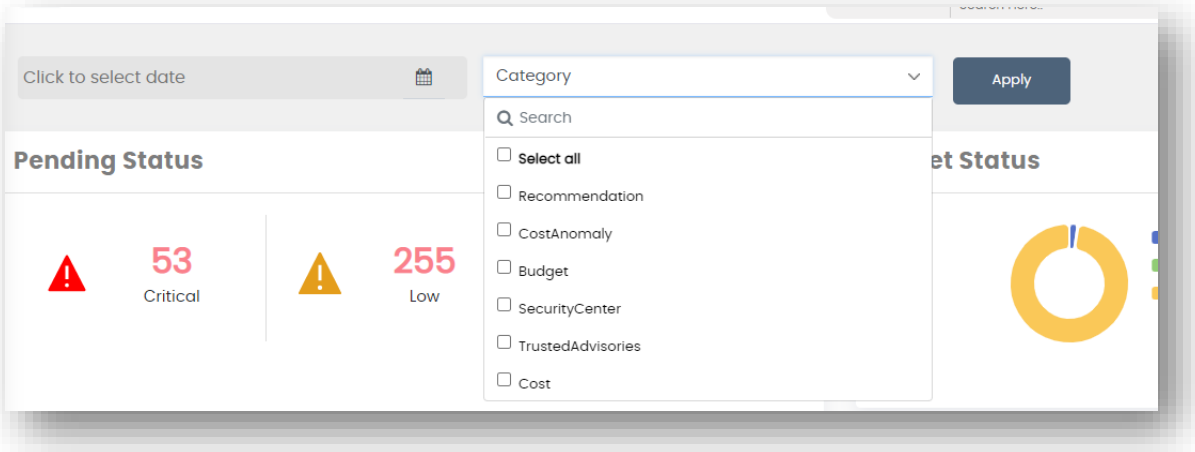

Figure 34 – Alert Dashboard (Cont.)

Alert listing:

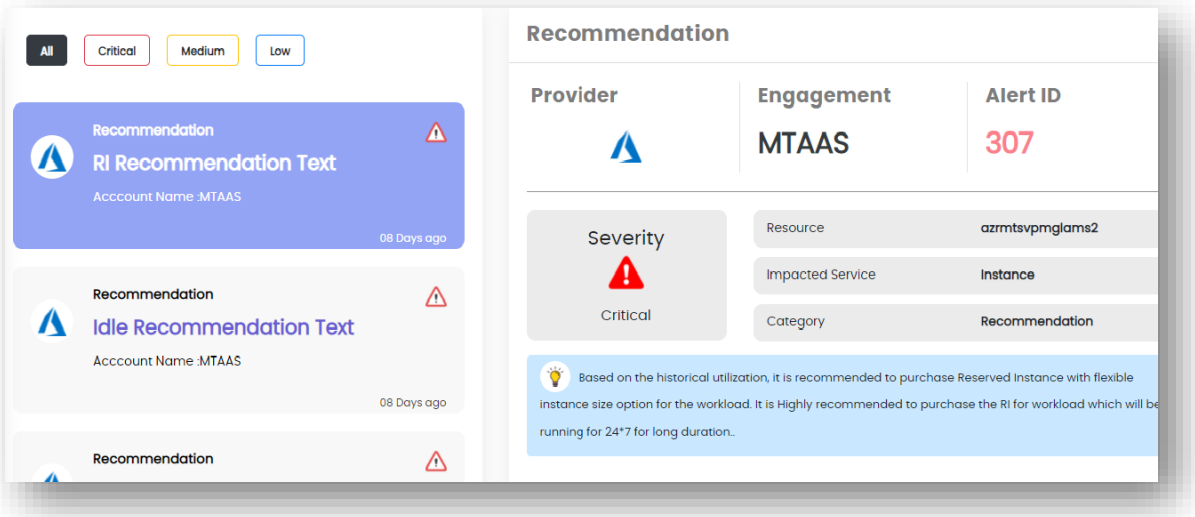

Figure 35 – Alert Dashboard (Cont.)

a. By clicking on any alert on the left side, the details for that alert can be seen on the right-hand side as shown in the above figure.

b. In the detailed view the alert can be assigned to any assignee putting some remarks.

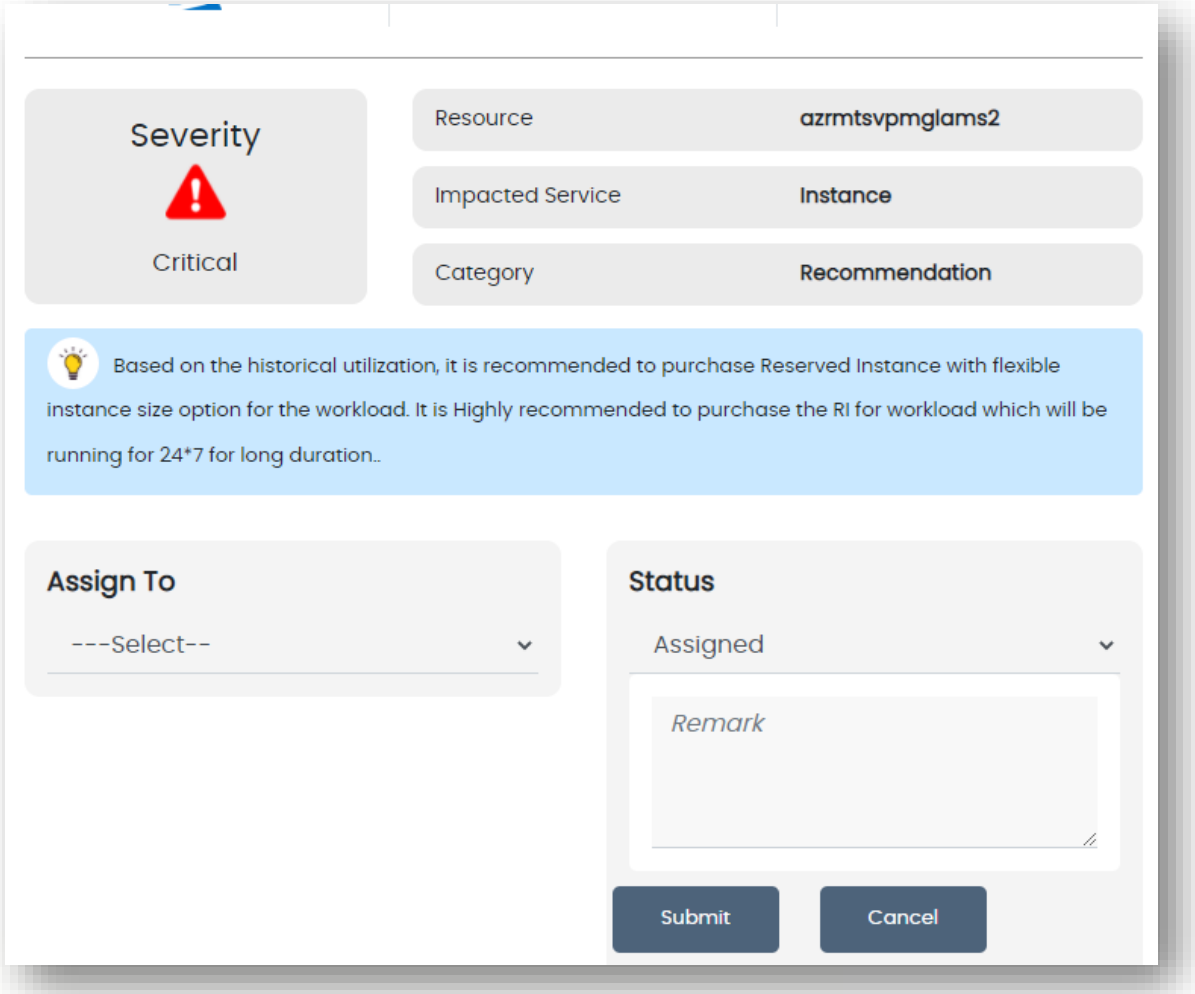

Figure 36 – Alert Dashboard (Cont.)

c. In the detailed view, there is a link View Work logs. This can be used to view the work logs done till date on the alert.

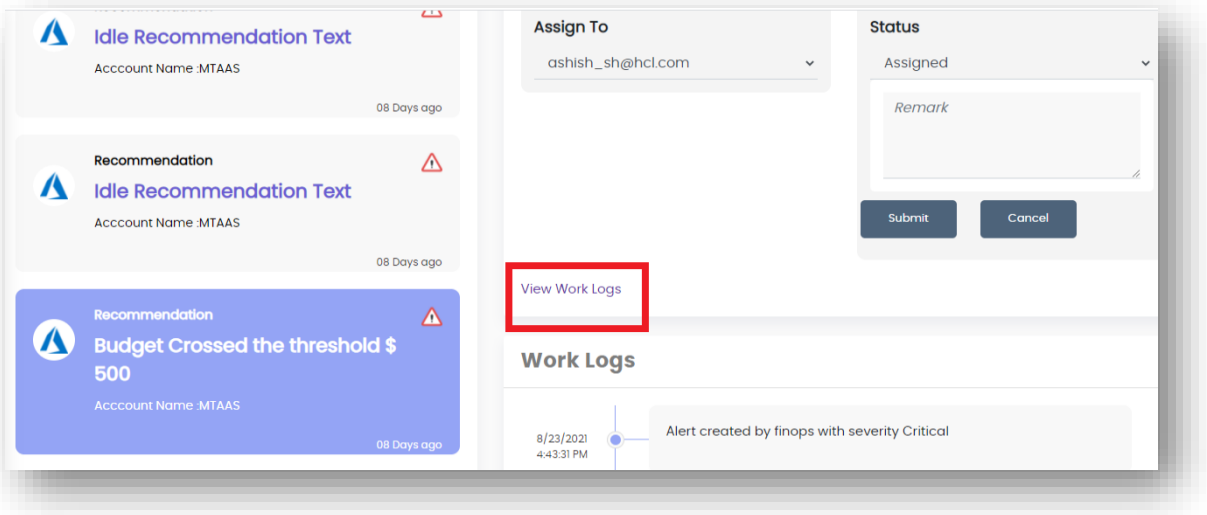

Figure 37 – Alert Dashboard (Cont.)

4.4 Alert Rule

This module helps in configuring rules for Cost Anomaly, Budget and Cost for generating alerts and sending emails based on defined ranges**.**

Rules can be defined for:

- Budgets
- Cost Anomaly
- Cost

Steps to configure the rule:

1. Navigate to Menu -> Configuration -> FinOps -> Alert Rule:

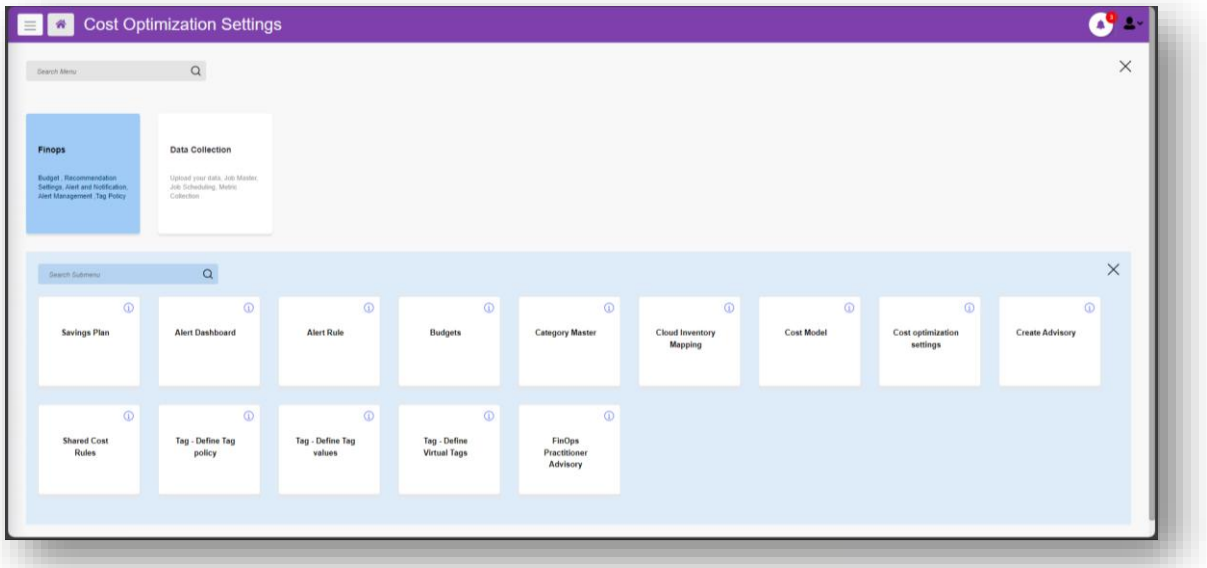

Figure 38 – Alert Rule

2. Click on Create Alert button on top left

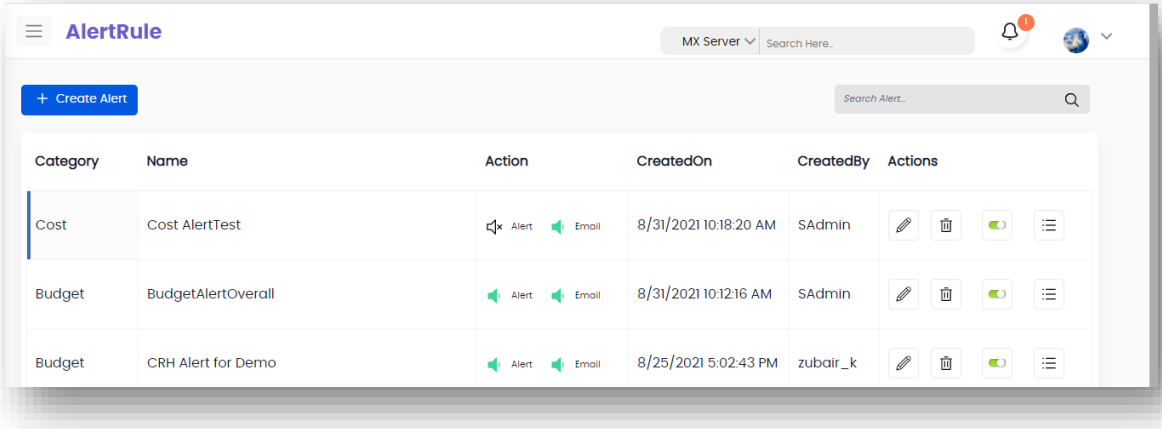

Figure 39 – Alert Rule (Cont.)

3. In the below page, following fields should be filled in:

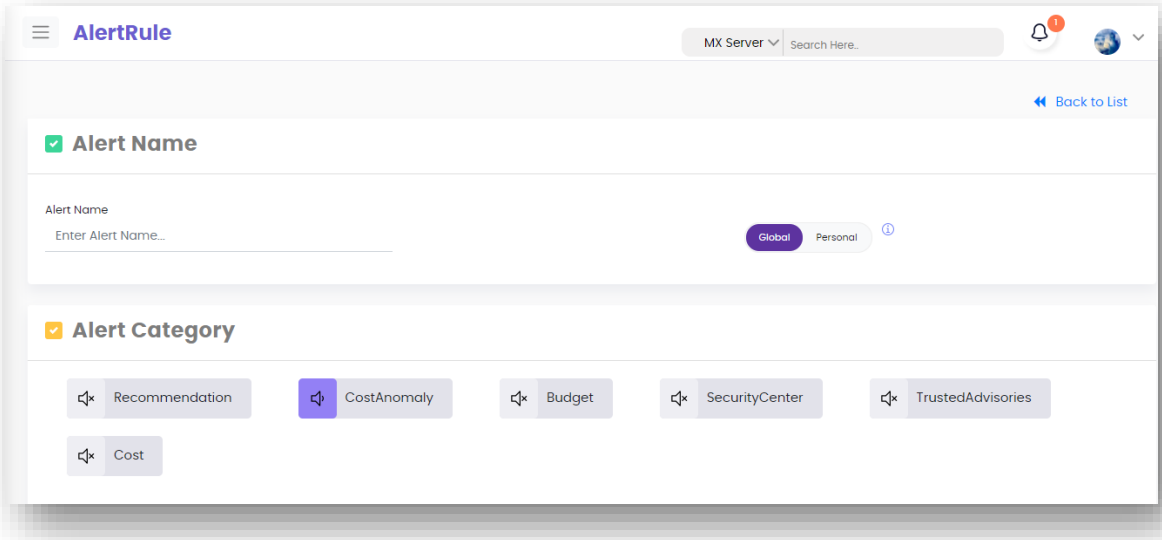

Figure 40 – Alert Rule (Cont.)

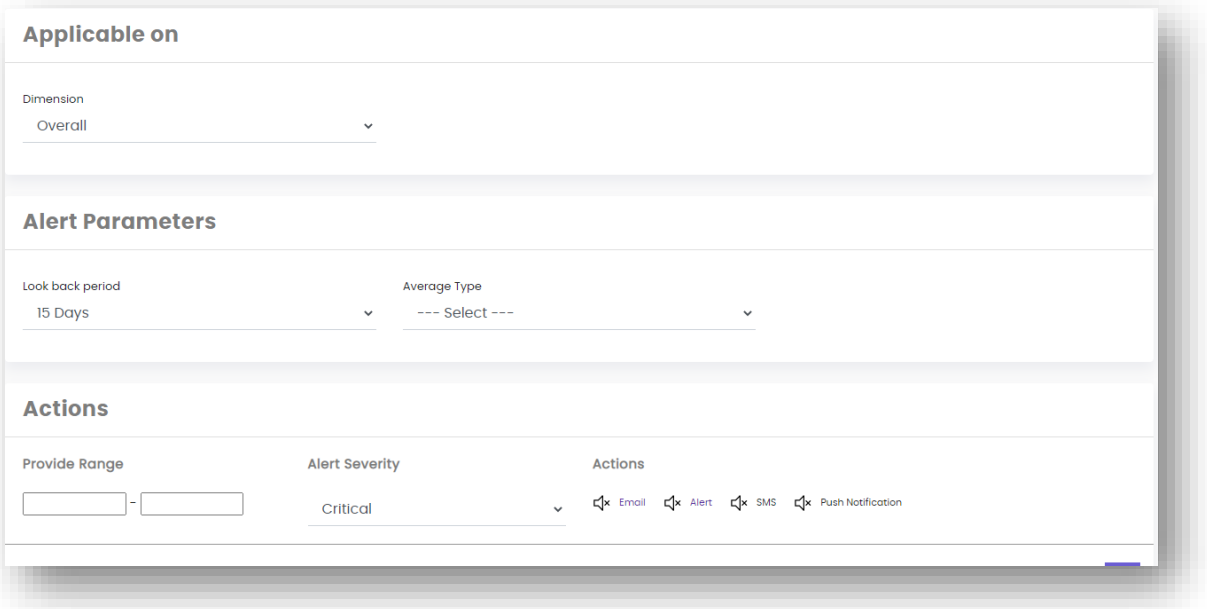

Figure 41 – Alert Rule (Cont.)

a. **Alert Name**: Enter a name for the alert rule

b. **Global/Personal**: Global rules can be configured when user needs to send email and generate alerts , whereas Personal rules can be configured at User level where only emails can be sent to the configured email ids.

c. **Alert Category**: Configuration to be done for which category: Budget, Cost Anomaly or Cost

d. **Budget Name**: If the category is selected as Budget, users need to select the name of the Budget for the rule that needs to be created. The budget list will contain all the budgets created for the user's engagement logged in.

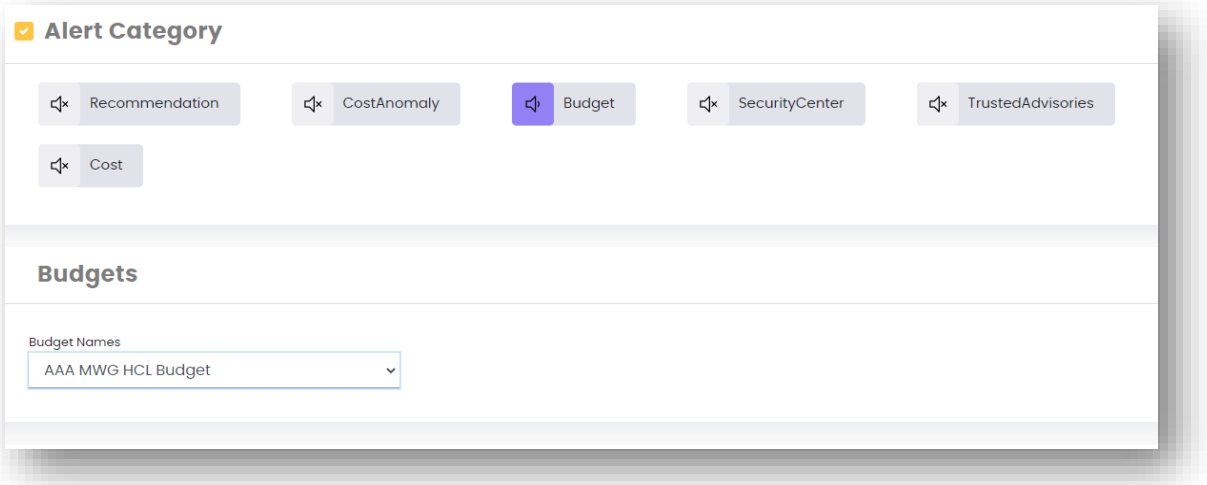

Figure 42 – Alert Category

e. **Applicable on**: Dimension for which the rule will be applicable. Options are Overall (for Customer), Subscription and Service. If user selects Subscription or Service, the list of Subscription/ Service will appear from which user can multi select

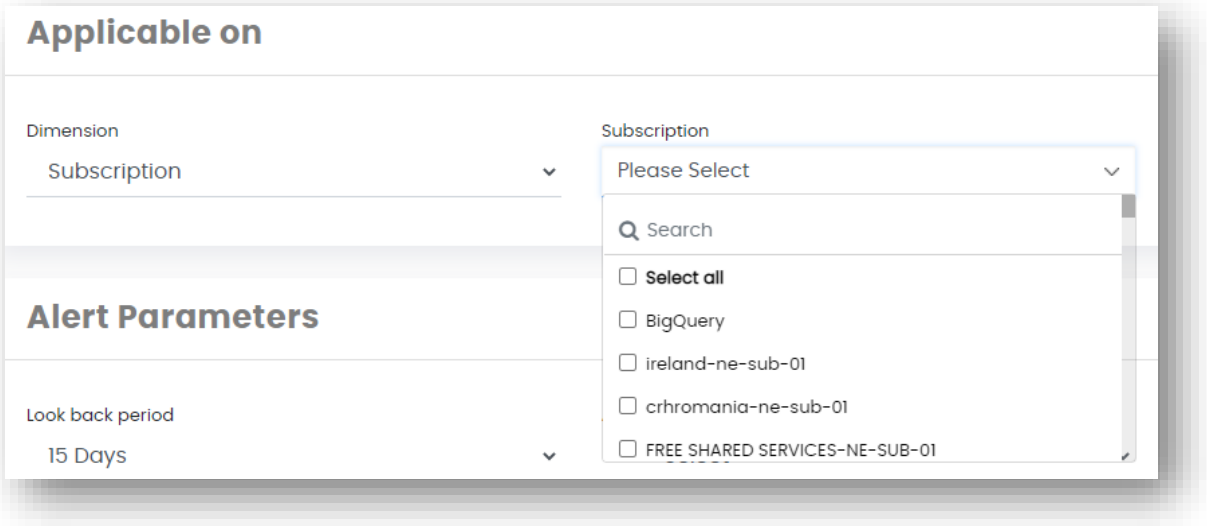

Figure 43 – Alert Category (Cont.)

4. If the category selected is Budget, then the dimension on which the selected budget is created will appear automatically. This dimension will not be editable.

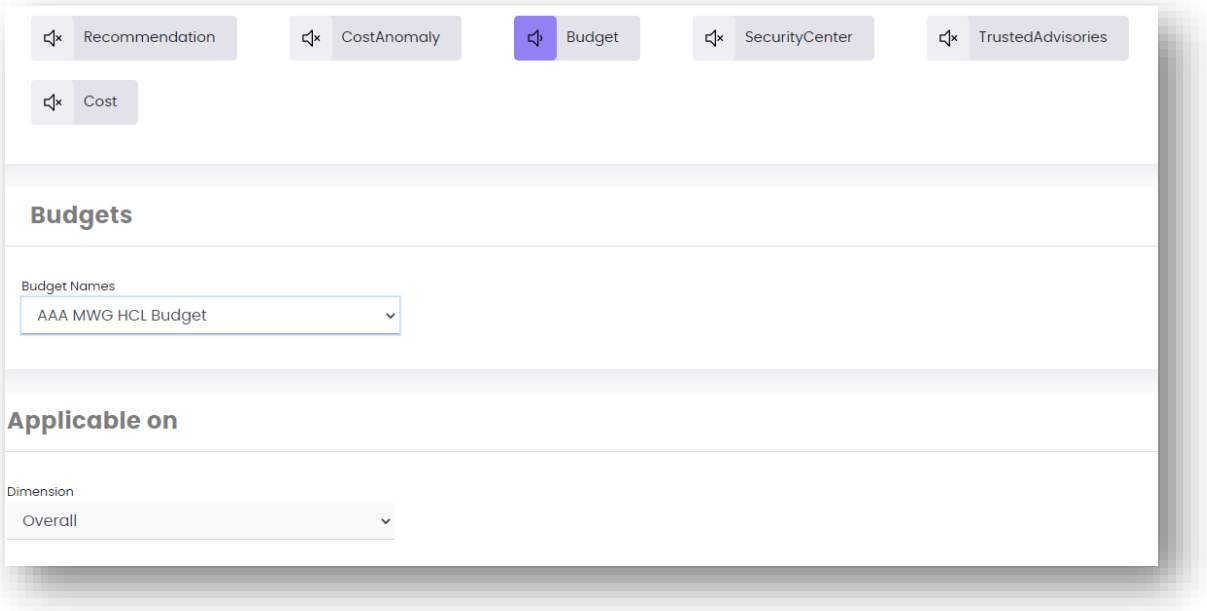

Figure 44 – Alert Category (Cont.)

5. Lookback Period: If the rule category is selected as Cost Anomaly user needs to mention the Lookback Period which will be considered for calculating daily average cost against which current anomaly will be calculated.

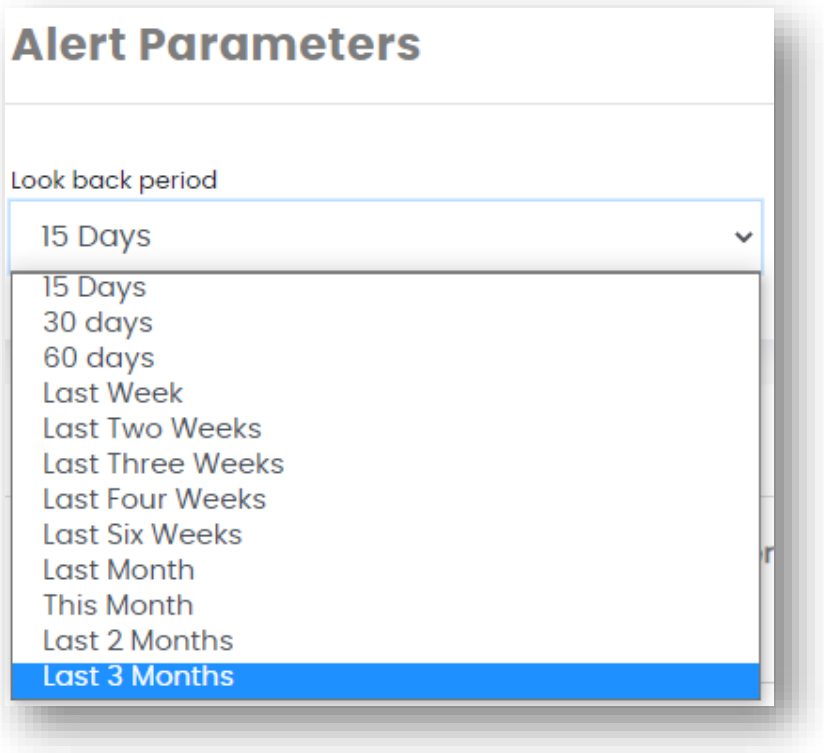

Figure 45 – Alert Category (Cont.)

6. Average Type: If the category is selected as Cost Anomaly or Cost, user needs to mention the period for which average cost to be considered for further calculation.

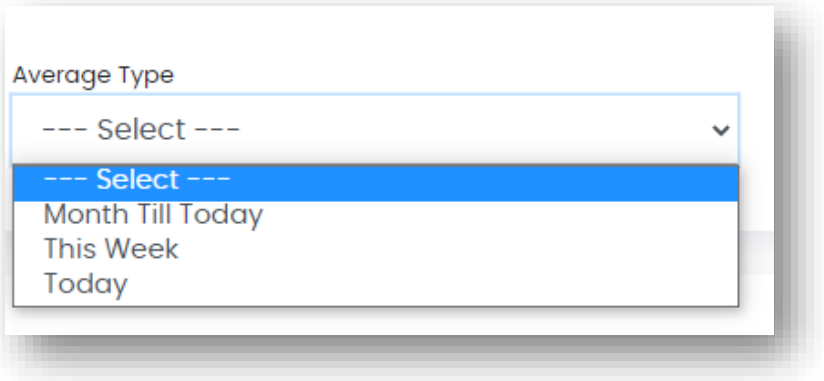

Figure 46 – Alert Category (Cont.)

7. Actions: User needs to define the ranges and severity for which he wants the email or/and alerts.

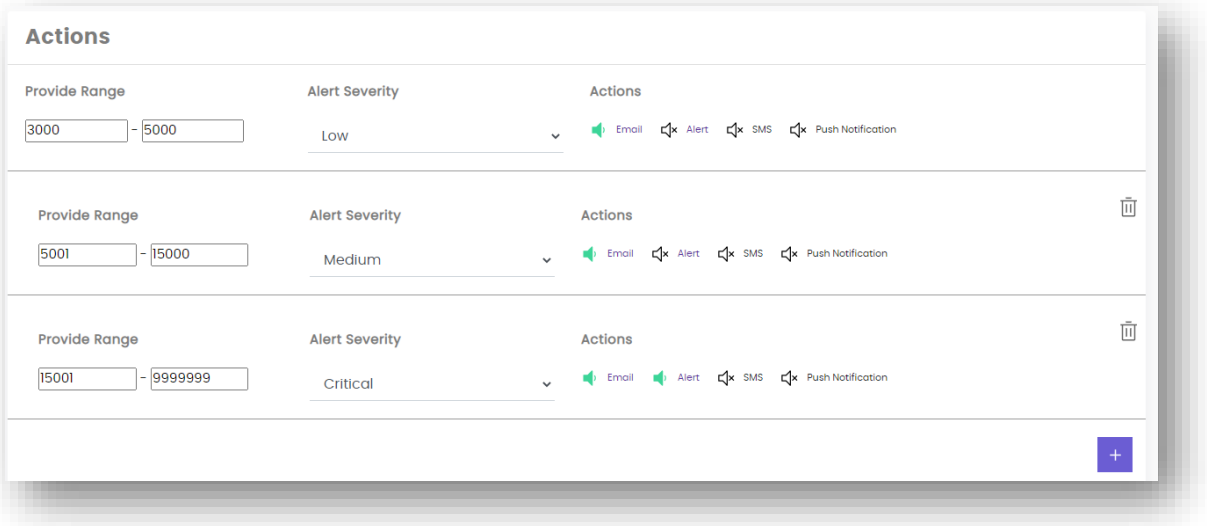

Figure 47 – Alert Category (Cont.)

8. The email ids to which mails should be sent should be configured by clicking on Email/ Alert links under Actions.

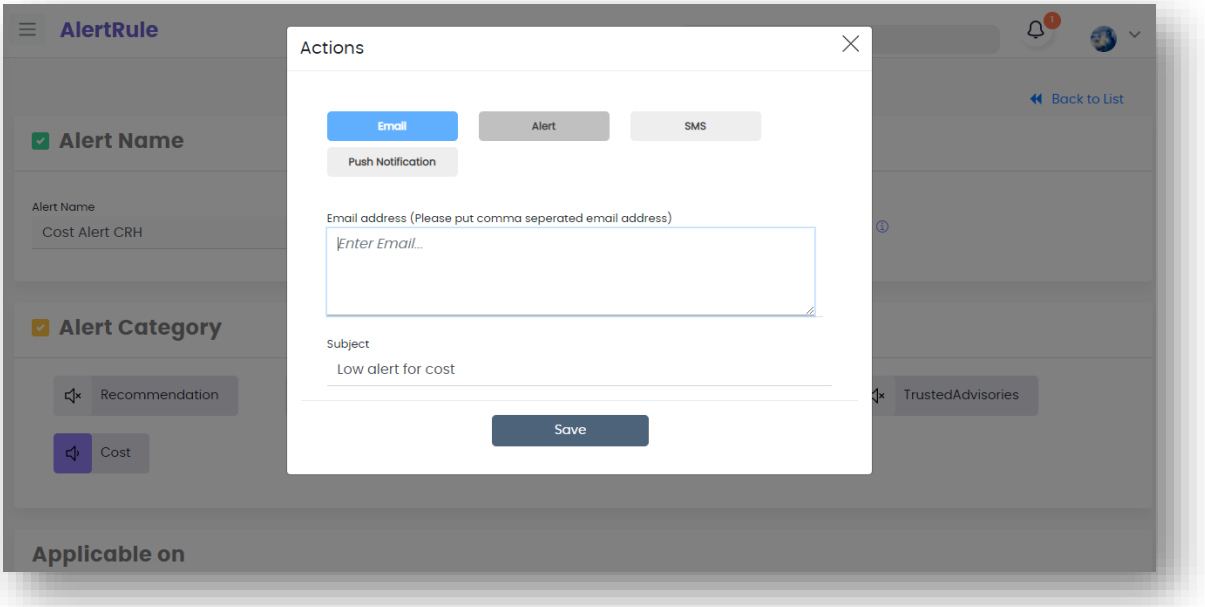

Figure 48 – Alert Rule (Actions)

9. Click on the Save button to save the configuration.

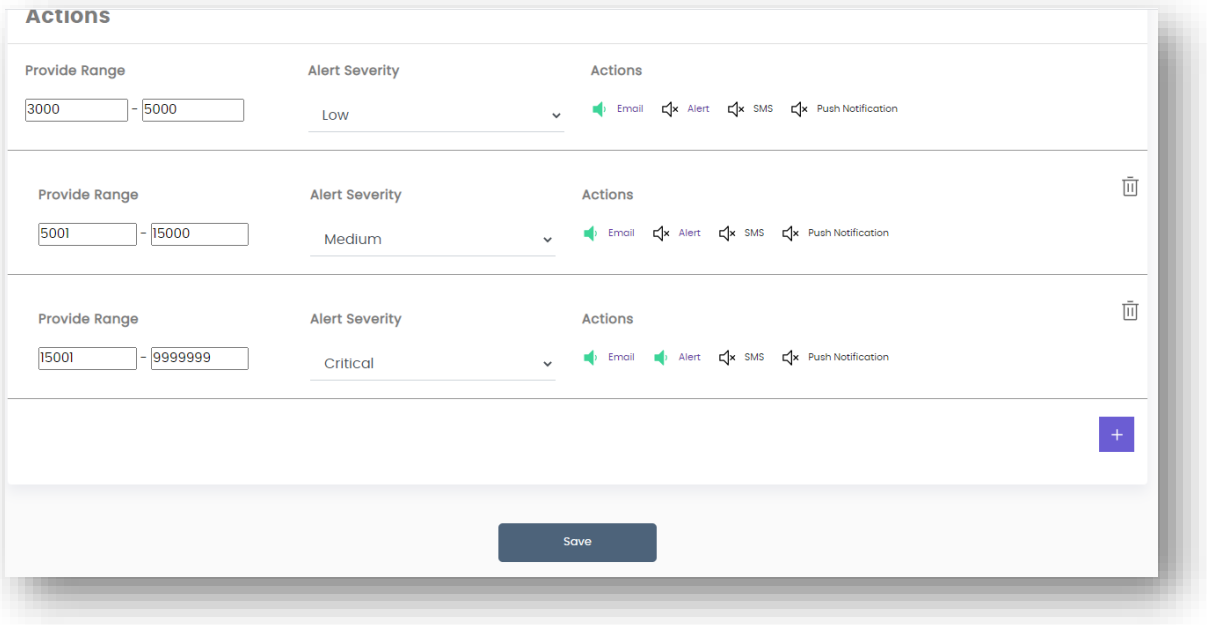

Figure 49 – Alert Rule (Actions) Cont.

10. The listing page enables user to Edit, Delete and Enable/ Disable a rule.

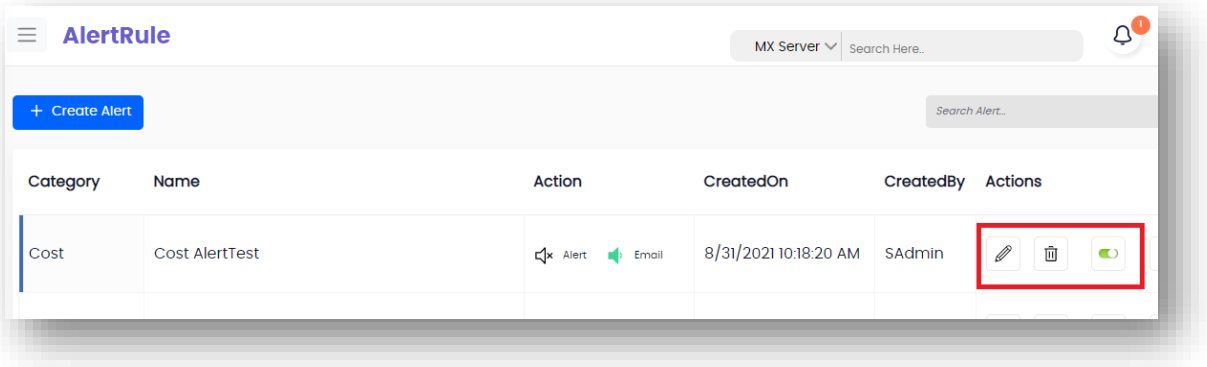

Figure 50 – Alert Rule (Cont.)

Here we can see different tabs technique wise. Each tab shows savings per resource. These can be assigned to assignees with severity. As soon as a line item is assigned to someone, an alert is generated. The alerts can be seen on Alert Dashboards.

4.5 Discount

Discounts can be configured for an engagement based on several dimensions. Dimensions precedence will override the discounts. To open this console, please navigate to -

- **Menu ->** Configuration -> FinOps -> Discount

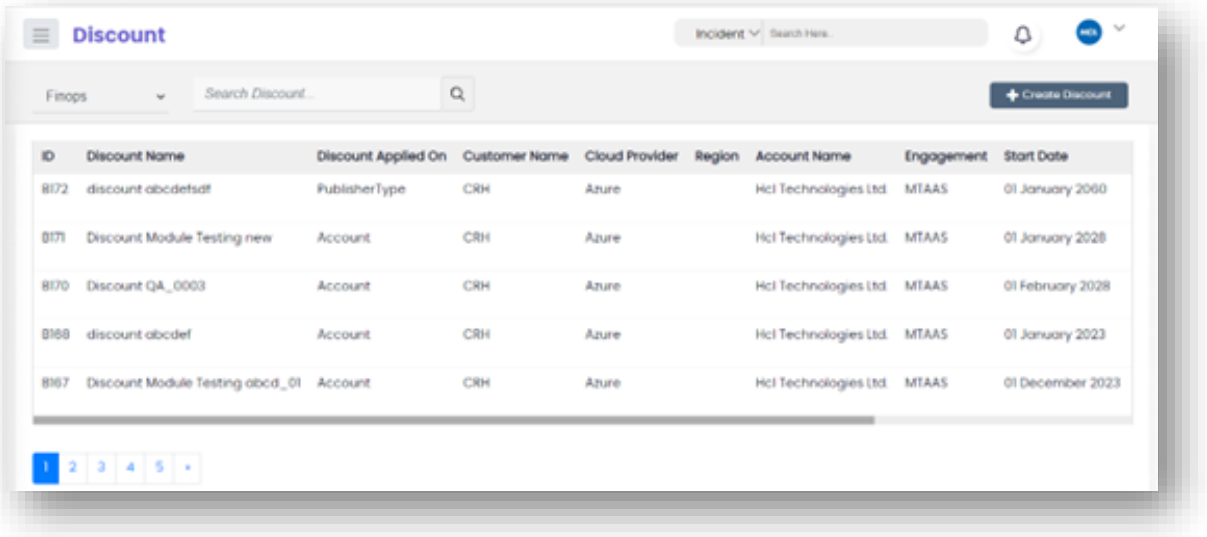

Figure 51 – Discounts

4.5.1 Create Discount

1. Navigate to Menu -> Configuration -> FinOps -> Discount -> click on Create Discount as shown below:

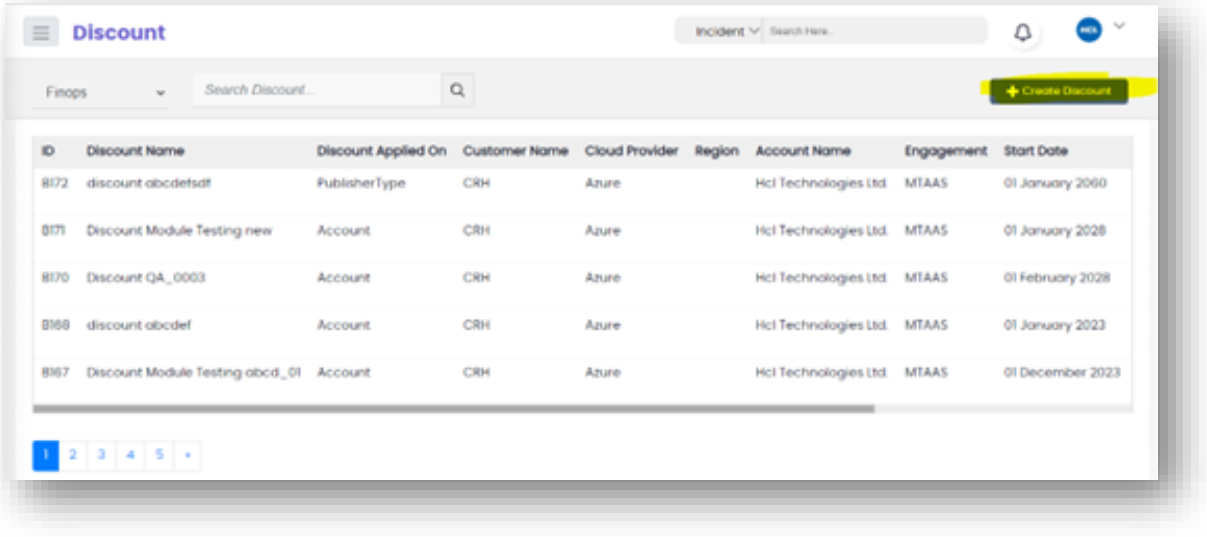

Figure 52 – Create Discount

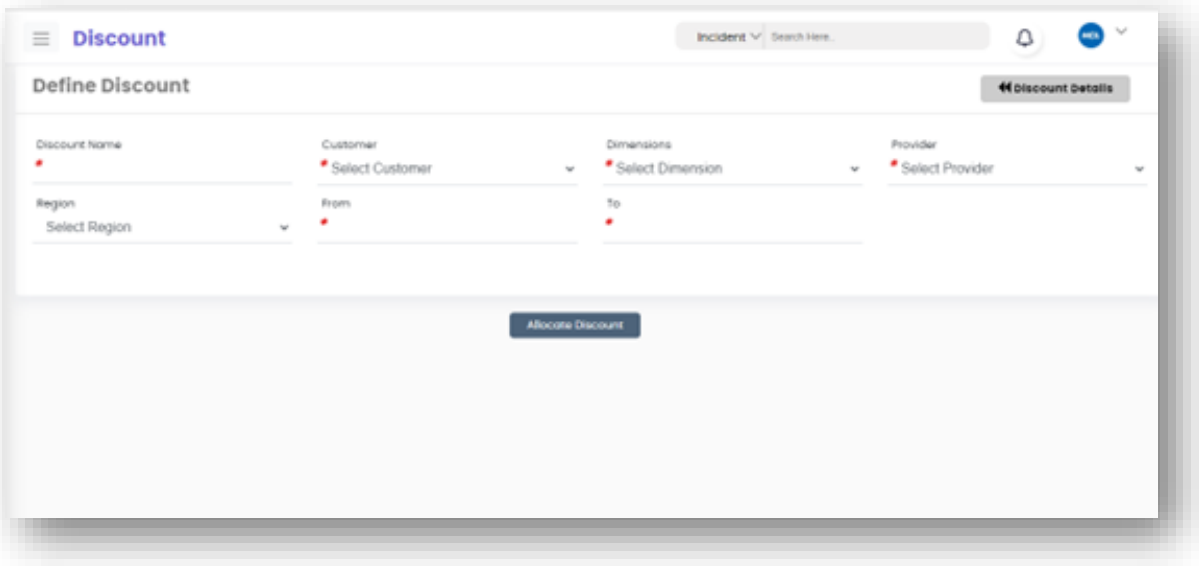

Figure 53 – Define Discount

2. Enter the Discount Name, select the customer from Customer dropdown, select the Dimension from Dimension dropdown, select the provider from Provider dropdown, select the Region from Region dropdown, select From and To Dates.
3. Then Press on Allocate Discount. It will show the Overall Discount details below:

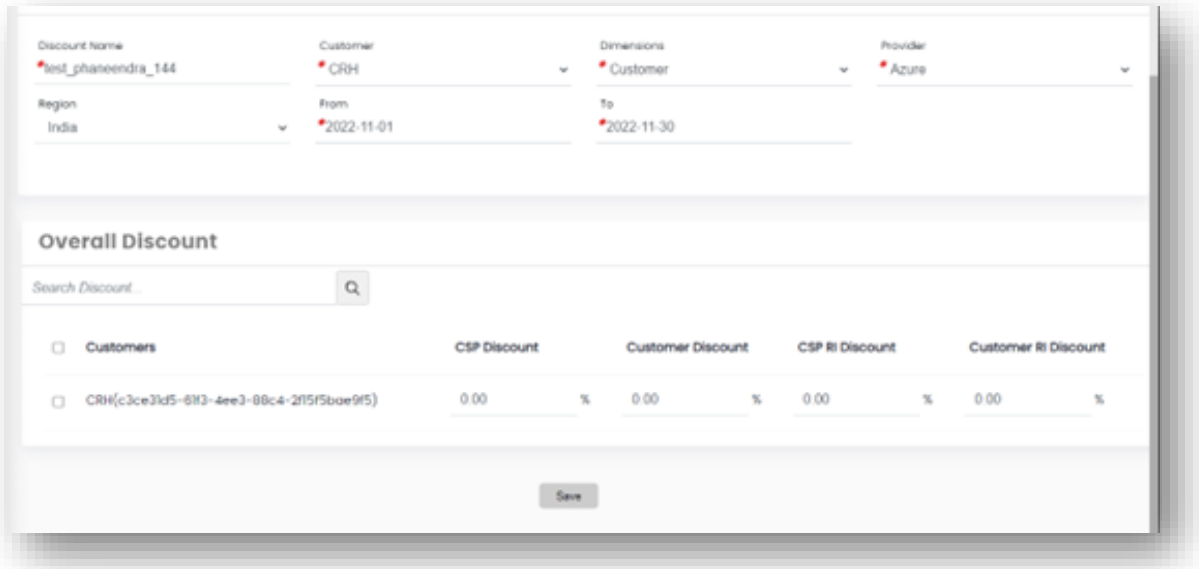

Figure 54 – Create Discount

4. Enter the Discount percentage for the Customers and select the checkbox.
5. Then click on ‘Save’ button to Save the Discount.

4.5.2 Edit Discount

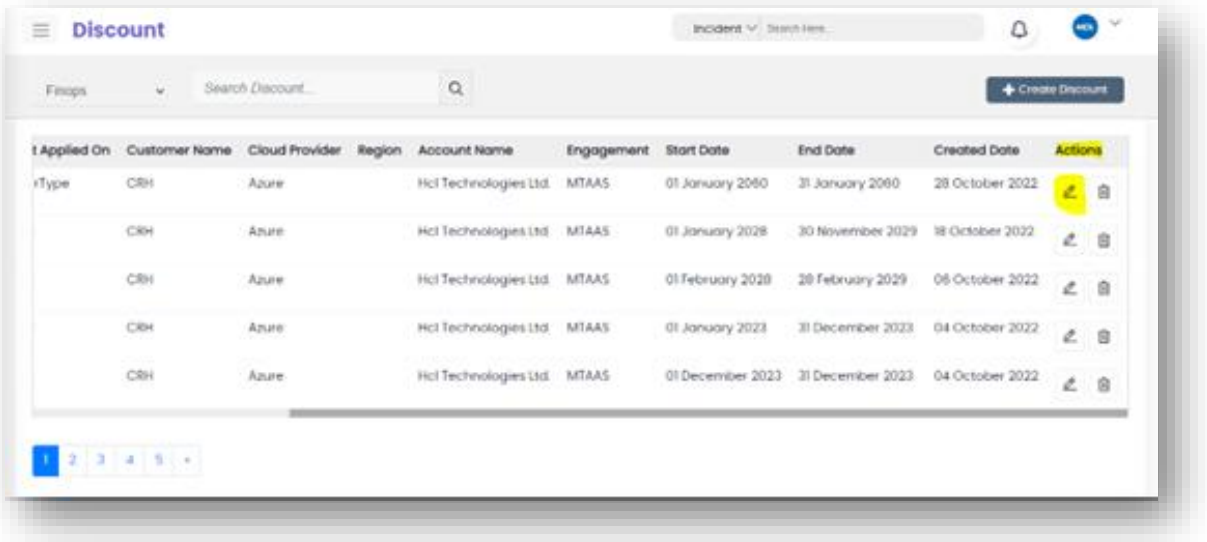

Figure 55 – Edit Discount

1. In the Discount List Page, for every discount there is an Action Column as shown above.
2. Click on Edit button, it will take you to Edit Discount page.

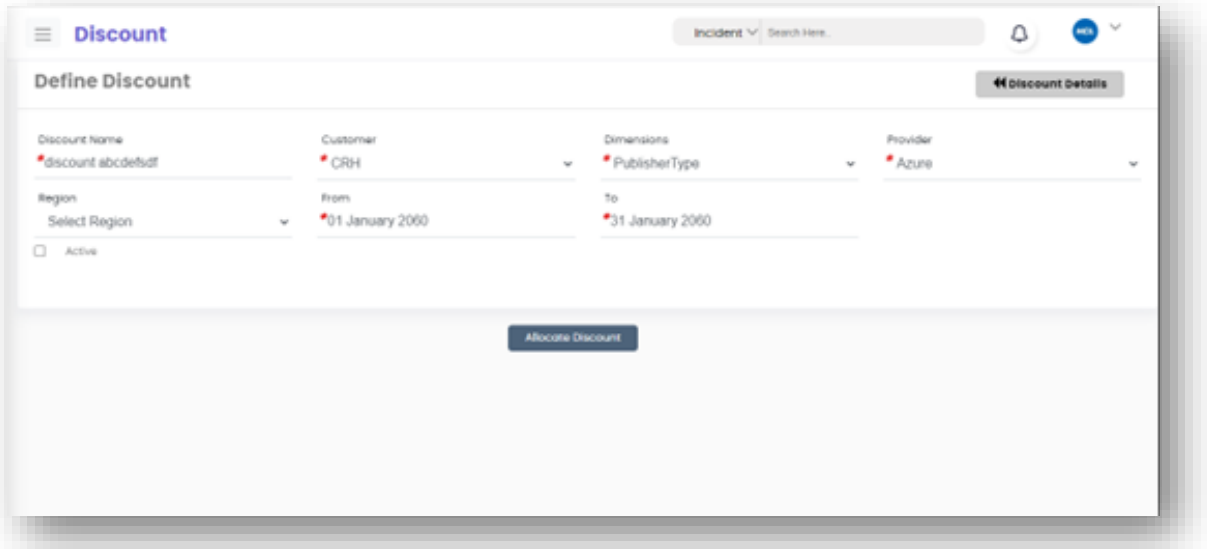

Figure 56 – Allocation Discount

3. Make changes which ever required and click on Allocate Discount. Then update the Discount percentage for the customers and click on Update button as shown below:

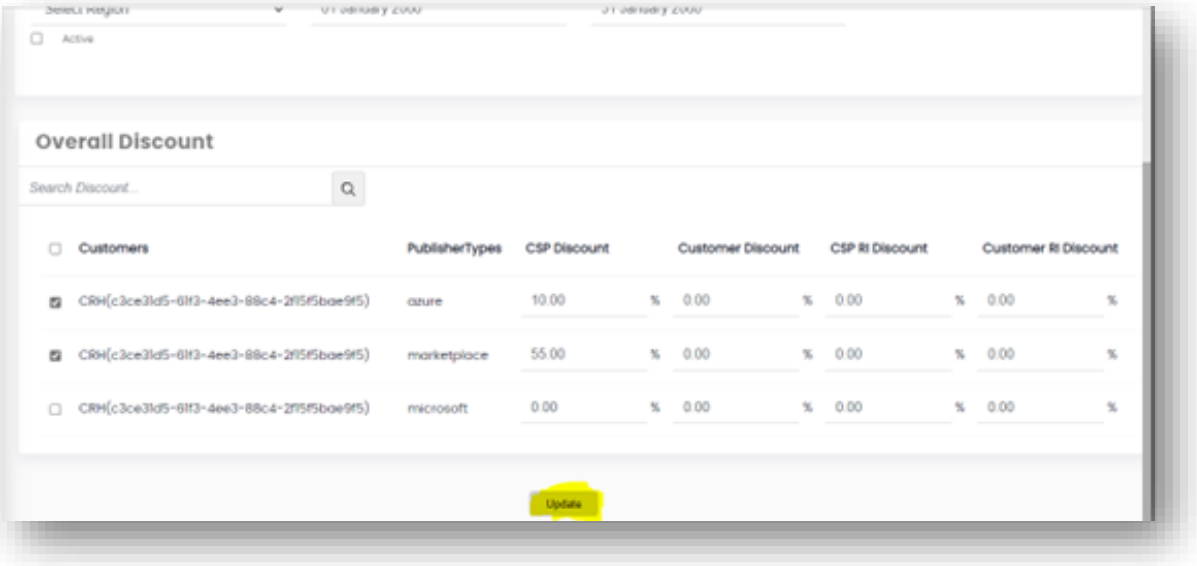

Figure 57 – Update Discount

4.5.3 Delete Discount

1. In the Discount List Page for every discount under the Action Column there is Delete button as shown above
2. Click on Delete button, to delete the Discount.

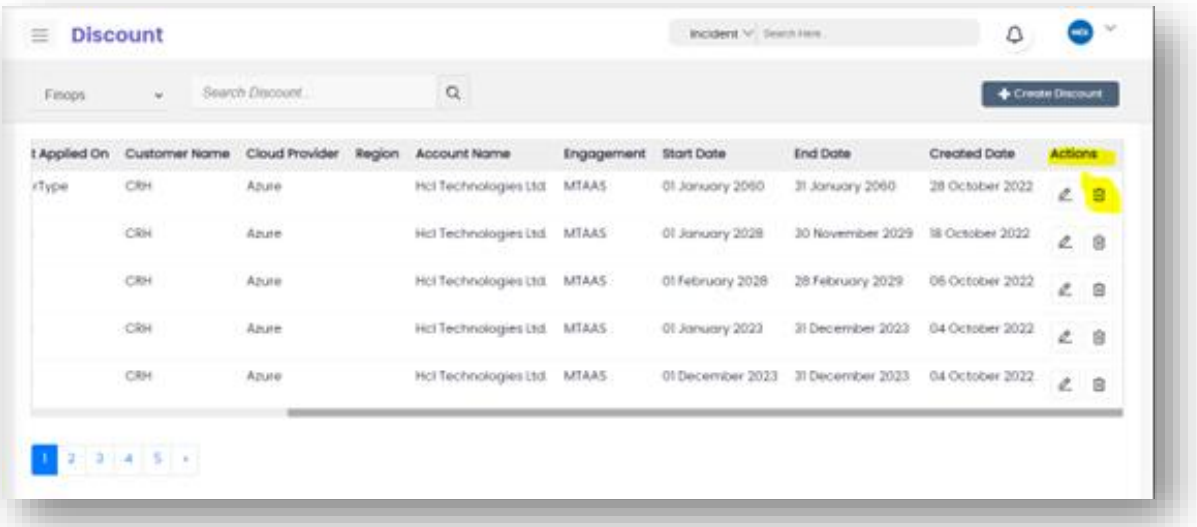

Figure 58 – Delete Discount

4.6 Creating CSP Customer Mapping

CSP Customer Mapping console can be accessed only by Super admin type of users. This console can be used to map CSP customers to an engagement. To open this page, navigate to -

1. Menu -> Configuration -> FinOps -> CSP Customer Mapping

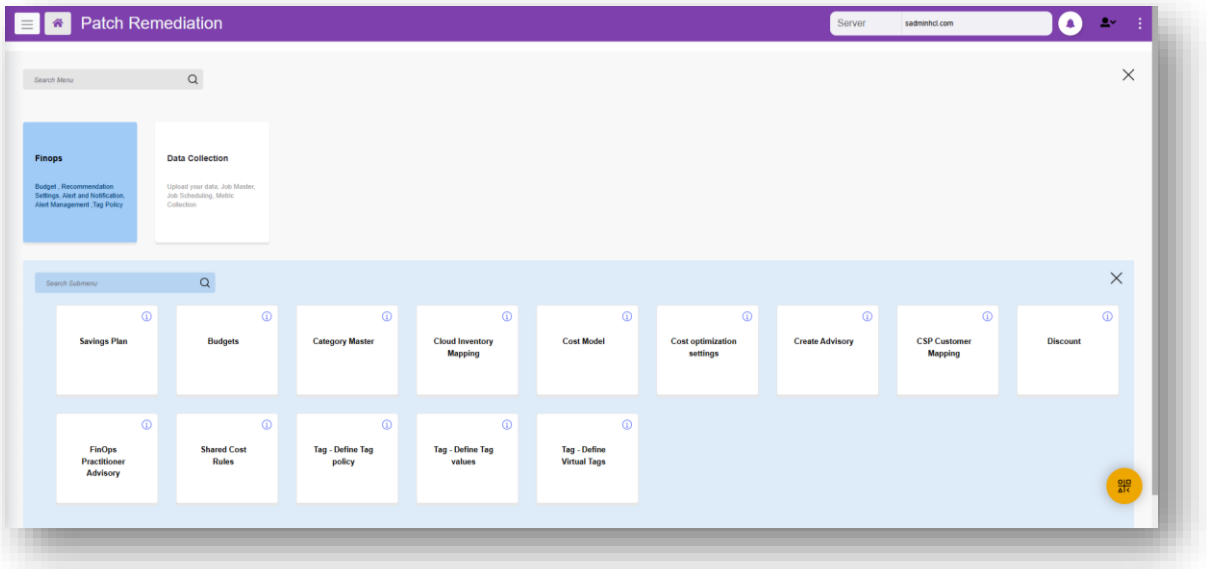

Figure 59 – CSP Customer Mapping

2. CSP Customer Mapping list page will open, then click on Create Customer Mapping as shown below.

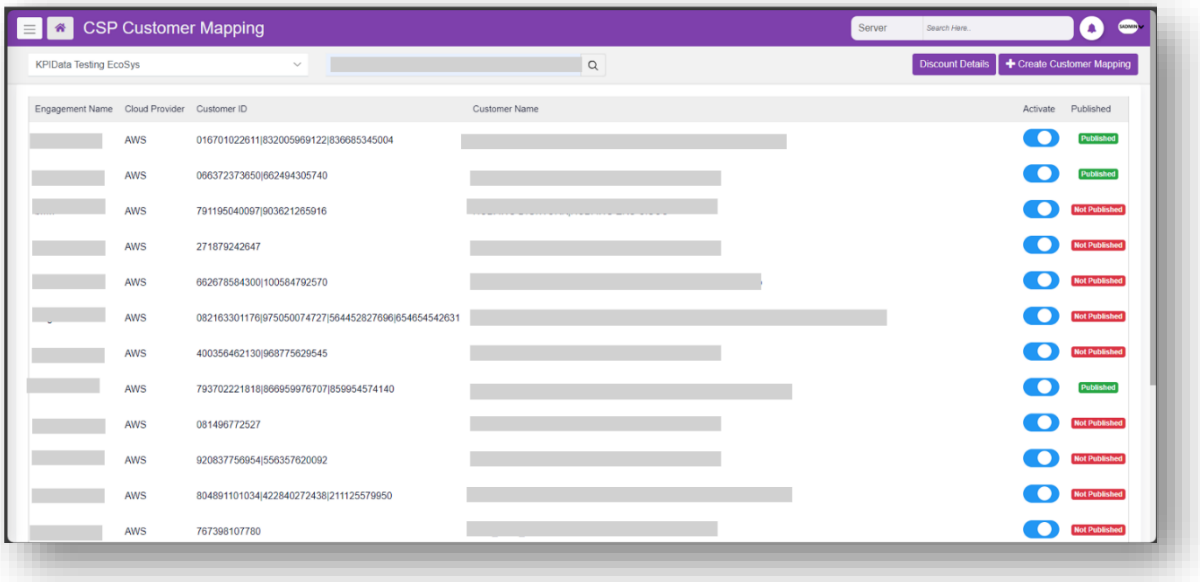

Figure 60 – CSP Customer Mapping

3. Here the Customer and Provider data will be populated from the CloudCustomers view. This console is mainly used to map CSP customers to an engagement. CSP Customers API updates data from CloudCustomers view.
4. Select Engagement, Provider and Customers (multi select) in the corresponding dropdown. And click Save button.

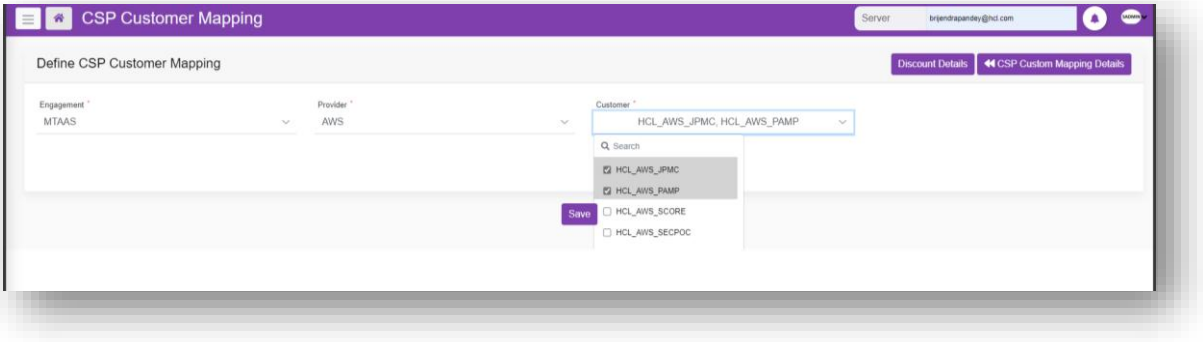

Figure 61 – CSP Customer Mapping (Cont.)

5. After saving You will be redirected to List page. By default, the created Mapping will be Active.

Use Cases:

If customers are already mapped to a particular Engagement, then you cannot map them to another Engagement**.**

4.7 Cloud Inventory Mapping (Cost Model)

Navigate to Menu -> Configuration -> FinOps -> Cloud Inventory Mapping to open this console.

Cloud Inventory List looks like the screen shot. Data will be visible hierarchical Cloud provider wise and then Version wise.

1. To Create new cloud inventory, click on the New Inventory Model button.
2. Edit- To Edit Record Use Edit icon. Users can Edit Cloud inventory model data until it is not activated once activated this option will not display in that record.
3. Disable the record use the disabled icon. Users can disable cloud inventory model data until it is not activated, and once activated this option will not display in that record.

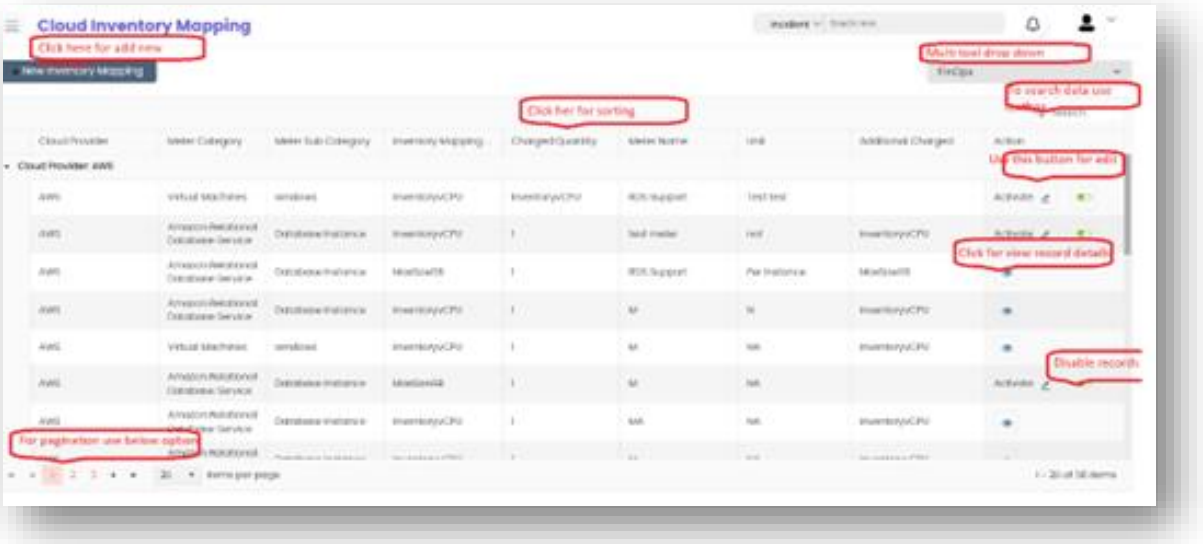

Figure 62 – Cloud Inventory Mapping

4. After clicking on view icon user can view cloud inventory details as below:

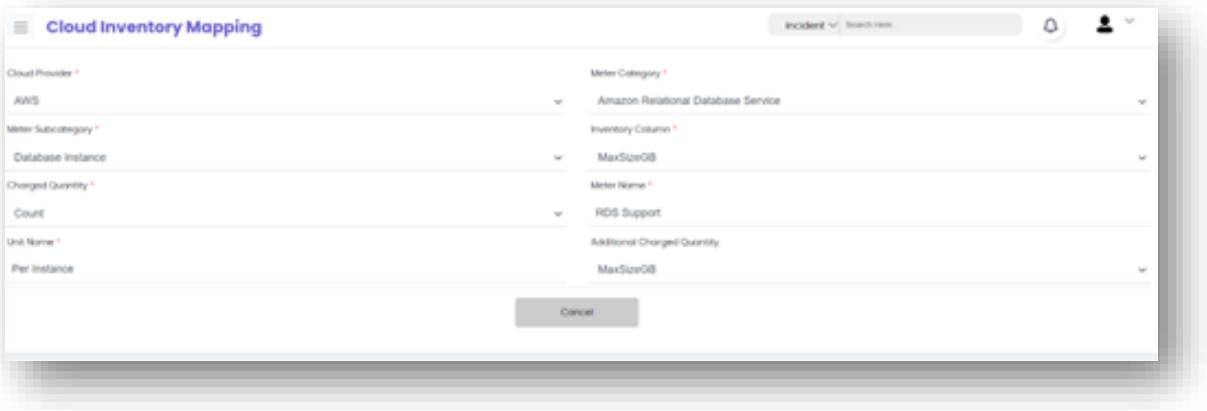

Figure 63 – Cloud Inventory Mapping (Cont.)

Section 1- Cloud Inventory:

```
select * from dbo.Cloud_invSubCategoryMapping with(nolock)
```

Table 2 – CostModelMaster

Column_name	Description
nmAutoID	unique id
ToolId	ToolID will be nmServiceSubTypeID from toolmaster
ObjectID	Object used for making entry for base or client
ToolId	ToolID will be nmServiceSubTypeID from toolmaster
MeterCategory	Meter category comes from Cloud_ComputeInventory_#EngagementID#
EngagementId	this is engagement id
CloudProvider	vcvalue as ID, vctext as [Name] from ParameterMaster where vcCode = 'CloudProvider' and chActive='Y'
MeterSubCategory	Meter subcategory comes from Cloud_ComputeInventory_#EngagementID# table based on meter category
TableName	It is a fixed table name based on EngagementId in backend processing Cloud_ComputeInventory_#EngagementId#
InvMappingCol	All returning column by this SP GetInventoryColumn @EngagementId
ChargedQuantity	Charged Quantity either 1 or column name returning by SP GetInventoryColumn @EngagementId
AddColInChargedQuantity	this is optional
MeterName	Meter Name provided by user
Unit	Unit provided by user
IsActive	
CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
bCustom	
EngagementId	This is engagement id
IsActivated	This flag used for making Inventory record activated if value contained true then user cannot modify details

- Cloud Provider is binding from the object below:

```
SELECT vcvalue as ID,  vctext as [Name] from ParameterMaster
where vcCode = 'CloudProvider'
Procedure - exec GetCostModelCloudProvider
@cloudProvider=NULL,@engagmentId=N'***',@toolId=N'***'
```

- For binding Meter category from below procedure:

```
exec GetMeterListByCloudProvider
@cloudProvider=N'AWS',@toolId=N'****'
```

- For binding Meter subcategory below stored proc using:

```
exec GetSubMeterCategory @MeterName=N'Amazon Relational
Database Service',@cloudProvider=N'AWS'
```

- For the binding Inventory column list, we are using the below stored procedures: -

```
exec GetMeterListByCloudProvider
@cloudProvider=N'AWS',@toolId=N'107'
```

- With Below stored procedure, we are binding all Cloud inventory data:

```
exec GetCloudInvSubCategroryMapping
MyDashboardProdMode=N'B',@EngagementId=N'***',@toolId=N'****',@
ID=NULL,@userId=N'****'
```

Category Master:

Category Master list looks like below screenshot. Data will be visible hierarchical Cloud provider wise and then Version wise.

1. To Create New Category Master, click on the Add Category Master button.
2. Edit- To Edit Record Use Edit icon. Users can Edit Category Master model data until it is not activated once activated this option will not display in that record.
3. Delete- To Delete the record, use the delete icon. Users can delete Category Master model data until it is activated once activated this option will not display in that record.

Figure 64 – Category Master

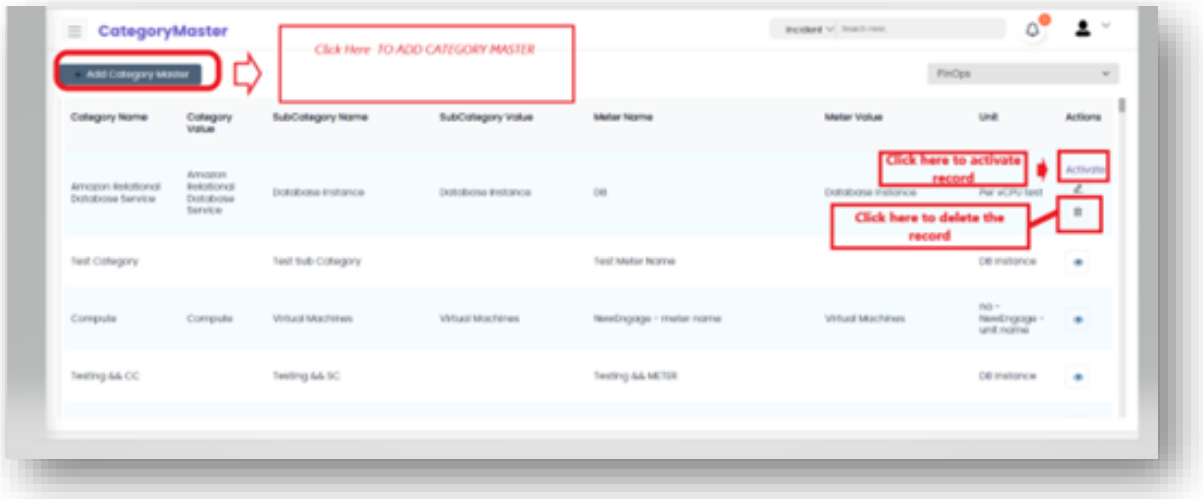

Figure 65 – Category Master (Cont.)

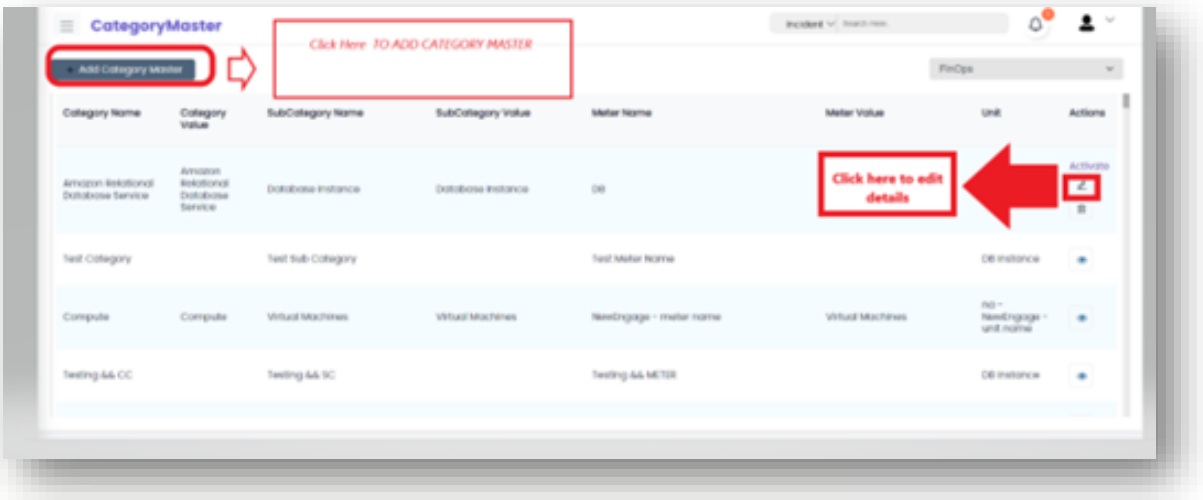

Figure 66 – Category Master (Cont.)

4. After clicking on view icon user can view category master details as below:

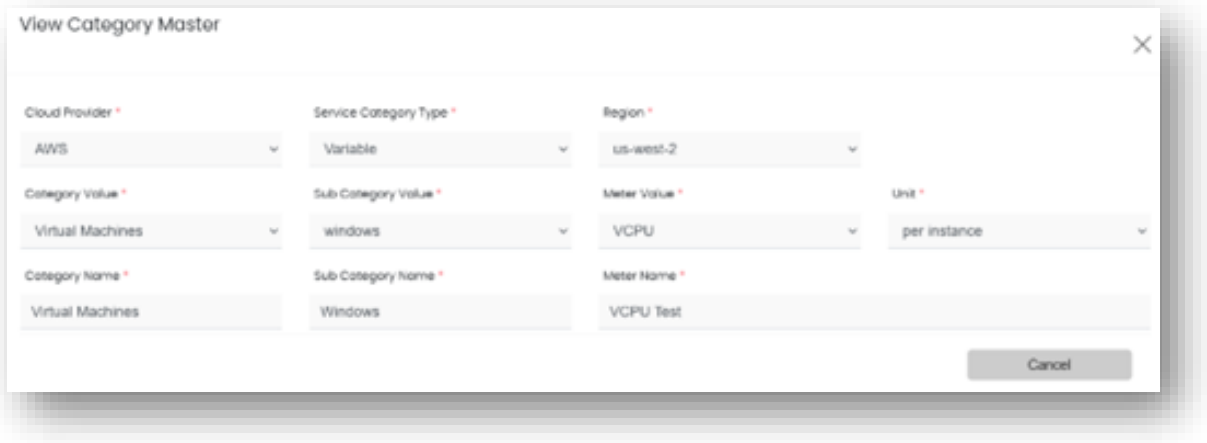

Figure 67 – Category Master (Cont.)

5. After clicking on edit icon user can edit category master details as below:

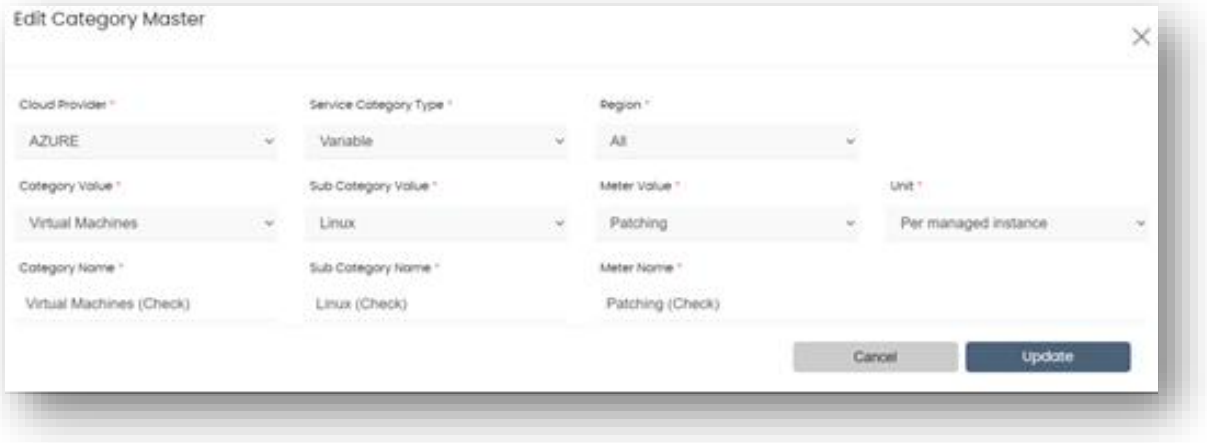

Figure 68 – Category Master (Cont.)

Table 3 – Category Master

Column_name	Description
CategoryName	Category Name Provided by user
SubCategoryName	Subcategory Name Provided by user
UnitId	Unit from VwCloud_invSubCategoryMapping view
CloudProvider	vcvalue from Select * from ParameterMaster where vcCode='Cloudprovider'
MeterCategory	Meter category comes from Cloud_ComputeInventory_#EngagementID# engagement wise
Description	Optional
ServiceCategoryTypeId	Either ‘Variable’ or ‘Fixed’
EngagementId	this is engagement id
Region	region id will come from mapping and region join
IsActive	This flag is used for making Category Master record delete if value contained true then record is deleted.

CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
isCustomCategory	Always true.
ToolId	ToolID will be nmServiceSubTypeID from toolmaster
bCustom	0 for dev, 1 for production environment.
Id	auto increment value
MeterName	Meter Name provided by user
CategoryValue	Meter category come from Cloud_ComputeInventory_#EngagementID table base on CloudProvider and EngagementWise
SubCategoryValue	Meter subcategory comes from Cloud_ComputeInventory_#EngagementID table based on meter category
MeterValue	MeterValue from vwCloud_invSubCategoryMapping
IsActivated	This flag is used for making Category Master record activated if value contained true then user cannot modify details

There are two types of Service Category: Variable and fixed.

- **Variable Service Category**: Variable Service Category we can create only when we have inventory data.
- **Fixed Service Category**: Fixed Service Category there is no need for inventory data we can use for fixed type cost.

Unit Dropdown will be filled based on Variable /Fixed Service Category:

If user select Fixed Service Type, then dropdown gets the value from Parameter Master like below Query:

```
Select * from Parameter Master where vcCode='Service Unit'
```

- Cloud Provider is binding from the object below:

```
SELECT vcvalue as ID, vctext as [Name] from ParameterMaster
where vcCode = 'CloudProvider'
Procedure - exec GetCostModelCloudProvider
@cloudProvider=NULL,@engagmentId=N'****',@toolId=N'****'
```

- For binding Meter category from below procedure:

```
exec GetCostModelServiceforinventory
ServiceCategoryType=N'Variable',@Serviceid=NULL,@CloudProvider=
N'AZURE',@EngagementId=N'****',@ServiceProvider=NULL,@Region=N'
All',@ToolId=N'****'
```

- For binding Meter Value from below procedure:

```
exec GetMeterFromInventorySubMapping @Serviceid=N'Virtual
Machines',@SubServiceid=N'Linux',@CloudProvider=N'AZURE',@Engag
```

```
ementId=N'****',@ServiceProvider=NULL,@Region=N'All',@ServiceCa
tegoryType=N'Variable',@ToolId=N'****'
```

- For binding Meter subcategory below stored proc using:

```
exec GetCostModelSubServiceforinventory @Serviceid=N'Virtual
Machines’,@SubServiceid=NULL,@CloudProvider=N'AZURE',@Engagemen
tId=N'****',@ServiceProvider=NULL,@Region=N'All',@ServiceCatego
ryType=N'Variable',@ToolId=N'***'
```

- For binding Unit below stored proc using:

```
exec GetUnitFromCloudCategoryUnitMapping @Serviceid=N'Virtual
Machines',@SubServiceid=N'Linux',@CloudProvider=N'AZURE',@Engag
ementId=N'****',@ServiceProvider=NULL,@Region=N'All',@ServiceCa
tegoryType=N'Variable',@MeterName=N'Patching',@ToolId=N'****'
```

- For the binding Region list we are using the below stored procedures:

```
exec  GetCostModelRegion @cloudProvider=N'AWS',
@engagmentId=N'****',@ToolId=N'****'
```

- For Save we are using below stored procedures:

```
Exec  [uspAddCloudCategory]
@MyDashboardProdMode=N'B',@LoggedinUserID=N'*****',
@EngagementID=N'****',@vcModelXML=N'<?xml version="1.0"
encoding="utf-16"?>
<ArrayOfSubServiceModel
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SubServiceModel>
    <IsActivated>false</IsActivated>
    <CategoryName>Amazon Relational Database
Service</CategoryName>
    <SubCategoryName>Database Instance</SubCategoryName>
    <CloudProvider>AWS</CloudProvider>
    <ServiceCategoryTypeId>Variable</ServiceCategoryTypeId>
    <Region>All</Region>
    <UnitId>Per vCPU  test </UnitId>
    <MeterName>CPU testt</MeterName>
    <isCustomCategory>true</isCustomCategory>
    <CategoryValue>Amazon Relational Database
Service</CategoryValue>
    <SubCategoryValue>Database Instance</SubCategoryValue>
    <MeterValue>CPU</MeterValue>
  </SubServiceModel>
```

```
</ArrayOfSubServiceModel>',@vcToolId=N'****',@Opertaion=N'I',@I
d=NULL
```

- Below stored procedure we are binding all CategoryMaster data:

```
exec  GetCategoryMasterList
@toolid=N'***',@EngagementID=N'****'
```

Cost Model:

Cost model List looks like the screen shot. Data will be visible hierarchical Cloud provider wise and then Version wise.

1. To Create new cost model, click on the New Cost Model button.
2. Edit- To Edit Record Use Edit icon. Users can Edit Cost model data until it is activated once activated this option will not display in that record.
3. Delete- To remove the record use the remove icon. Users can remove Cost model data until it is activated once activated this option will not display in that record.

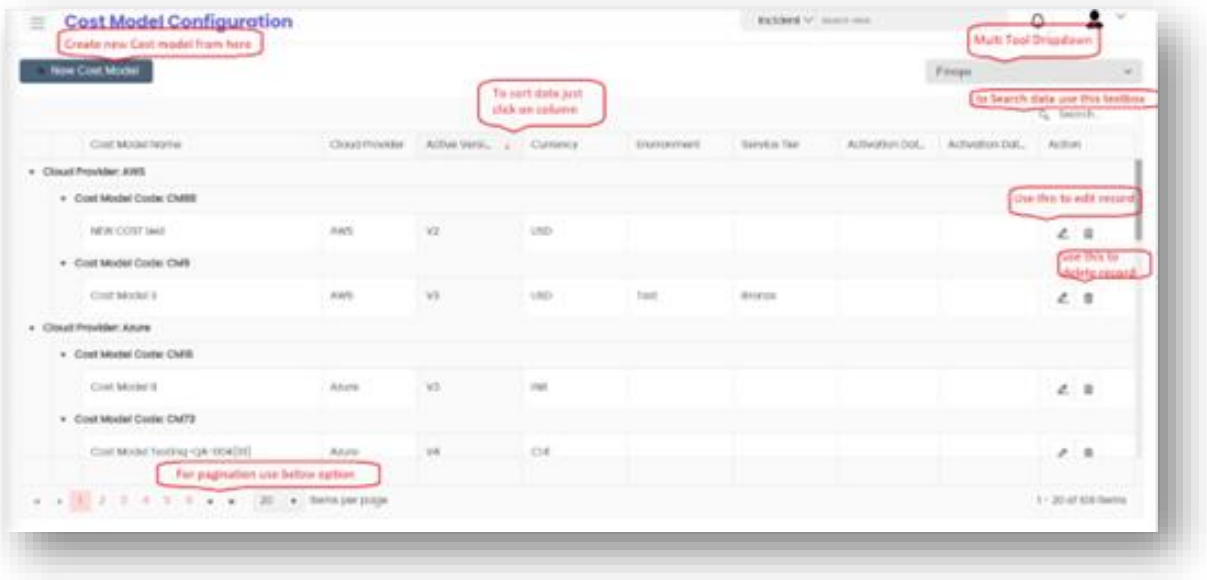

Figure 69 – Cost Model Configuration

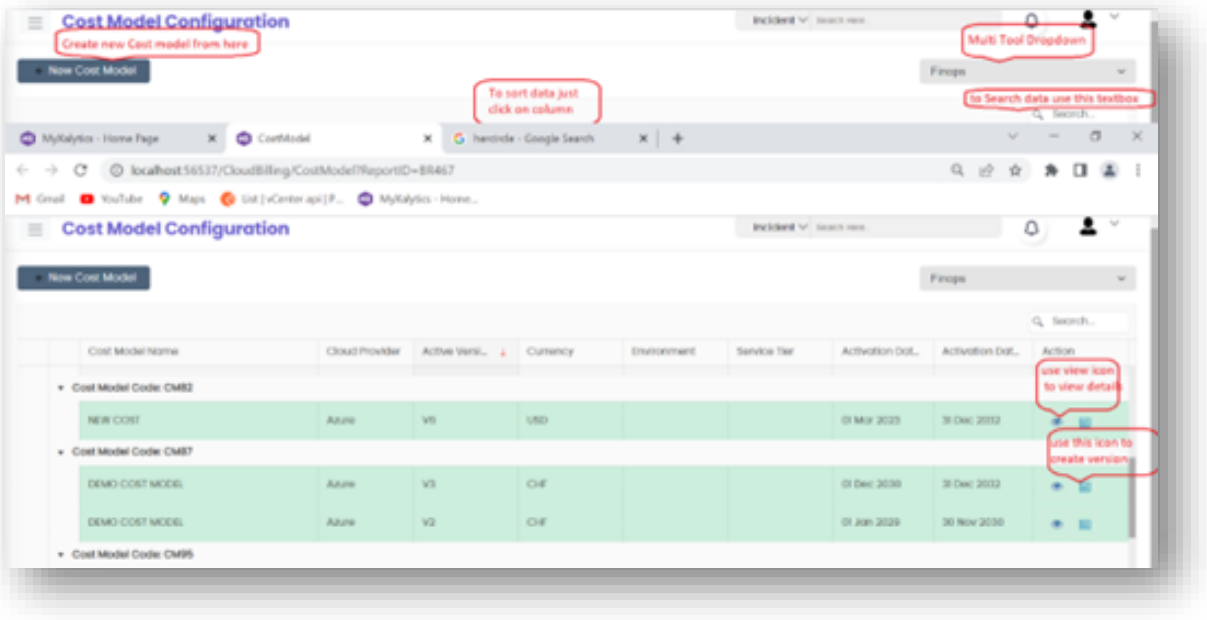

Figure 70 - Cost Model Configuration (Cont.)

4. If the cost model is activated, then row color will be green highlighted.
5. To view data user can click view icon.
6. If the user wants to change anything into an activated cost model, then the user can create a version of that cost model once the user will click version icon then user can display all data then he/she can create version by click save button from first section. Until the user clicks the save button version will not be created.

Section 1-Cost Model:

In this section we can add basic details:

Enter some basic details in this section. This data will be stored in the object below and one entry will be created in the version table after saving basic details:

- Select * from CostModelMaster
- Select * from CostModelVersionDetail

Table 4 – Cost Model Master

Column_name	Description
Id	Unique id
CostModelType	not in use
CostModelName	Cost Model name
ToolId	ToolID will be nmServiceSubTypeID from toolmaster
ServiceProviderId	Not in use
CustomerId	This is engagement id
CloudProviderId	vcvalue from Select * from ParameterMaster where vcCode='Cloudprovider'
DataCenterId	Not in use
RegionId	Region id will come from mapping and region join

LocationId	Location option will be in Vcenter provider which will bind from LocationMaster. exec GetCostModelLocation @RegionId=N'Asia-Pacific',@ToolId=N'***',@EngagementId=N'****'
CurrencyId	Currency id we save from Select * from ParameterMaster where vcCode='currency'
vcProfileId	This is a common property but not using in any filter
CostModelDescription	This is optional
IsActive	
CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
Startdate	not in use
DimensionId	not in use
bCustom	
VCenterId	Vcenter Name
ServiceTierId	Service id will be bind from Cloud_ComputeInventory_#Engagement# table
Environment	Environment will bind from Cloud_ComputeInventory_#Engagement# table

Table 5 – CostModelVersionDetail

Id	
CostModelId	Cost model id from costmodelmaster table
VersionNo	Version Number
ActivationDate	Activation start date
DeActivationDate	Activation end date
IsApplicableForExistingOrders	if the cost model applicable for backdate collection
IsActive	cost model will be active if its Y
CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
bCustom	
ToolId	
CustomerId	

- Cloud Provider is binding from the object below:

```
SELECT vcvalue as ID,  vctext as [Name] from ParameterMaster
where  vcCode = 'CloudProvider'
```

```
Procedure - exec GetCostModelCloudProvider
@cloudProvider=NULL,@engagmentId=N'****',@toolId=N'****'
```

- For regions binding below objects are used. object Cloud_CustomerRegionMapping is used for user friendly names:

```
Select * from Cloud_CustomerRegionMapping a with(nolock)
```

```
left join Region b with(nolock) on a.RegionCode=b.Code and
a.CloudProvider=b.Provider
```

```
Procedure - exec GetCostModelRegion
@cloudProvider=N'AWS',@engagmentId=N'***',@ToolId=N'****'
```

- To bind Currency below object is using:

```
Select * from ParameterMaster where vcCode='Currency'
```

- ServiceTier and Environment are binding from Cloud_ComputeInventory_#EngagementID#

```
Select ServiceTier,Environment from
Cloud_ComputeInventory_#EngagementID#
```

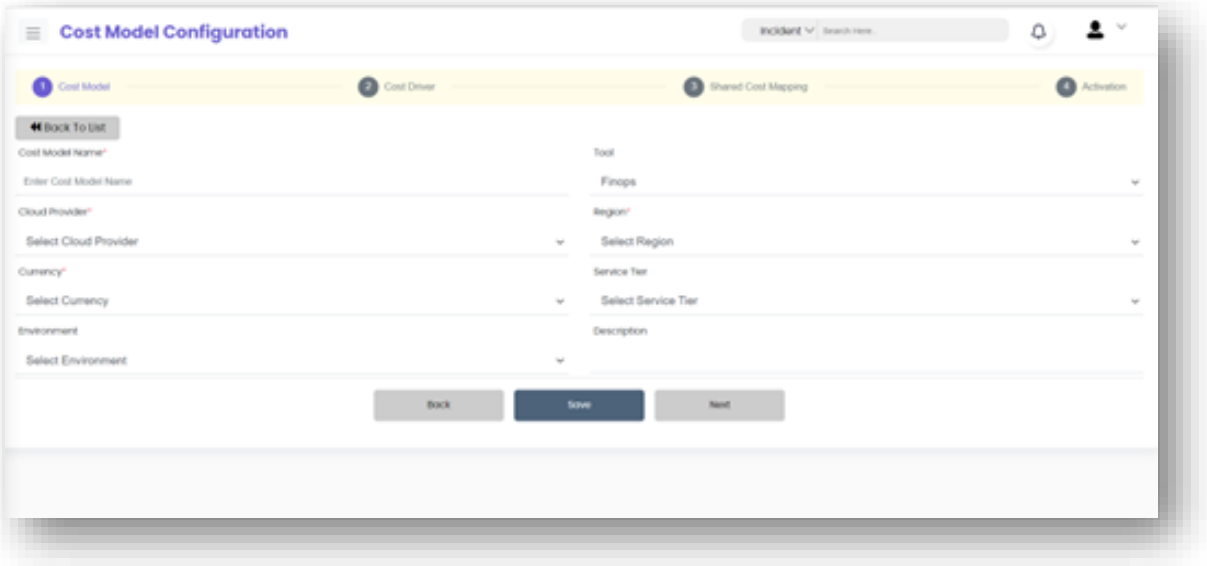

Figure 71 – Cost Model Configuration (Cont.)

Section 2-Cost Driver Section:

This section is not mandatory, so the user can skip this. Users can use this if Custom billed needs to be generated.

There are two sections where we can create Variable Cost and fixed Cost.

- Variable Cost line items: Variable line items we can create only when we have inventory data.
- Fixed Cost line items: Fixed line items there is no need for inventory data this section we can use for fixed line items.

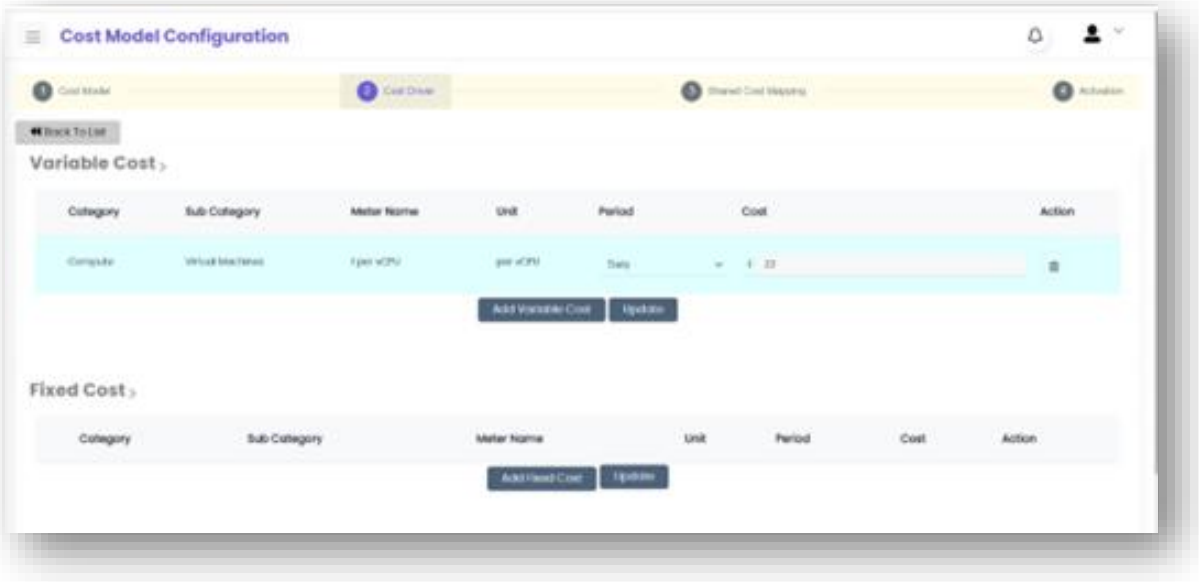

Figure 72 – Cost Model Configuration (Cont.)

Section 2.1-Variable Cost:

- Go to Variable Section and click on Variable cost button you will get a popup as mentioned below:
- If you want to create a new category, then you can navigate to "+ Category/Subcategory" button.

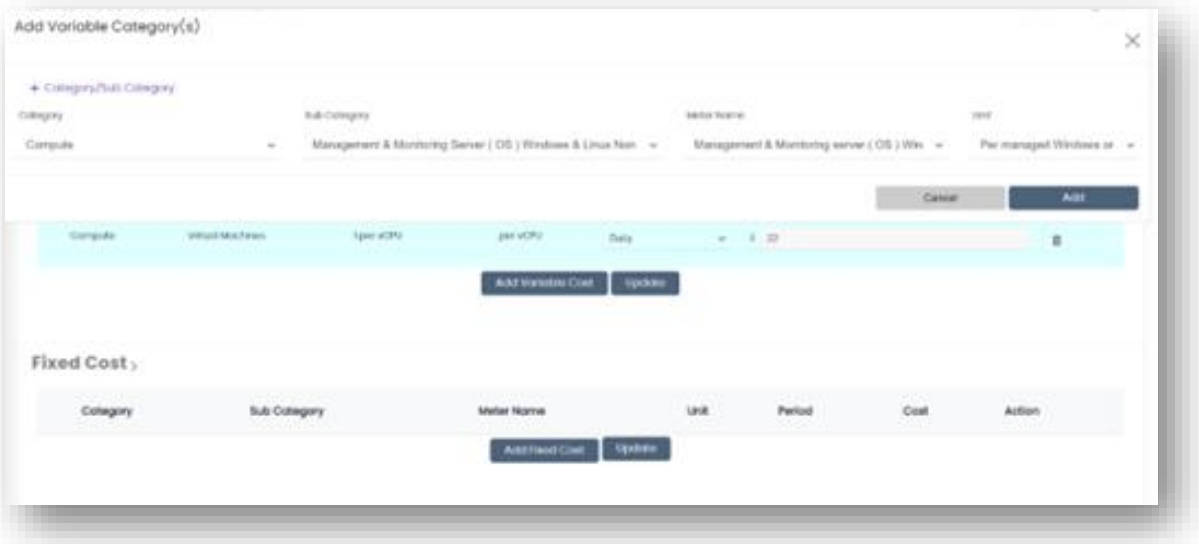

Figure 73 – Cost Model Configuration (Cont.)

Section 2.1. I - Add New Category in Variable Cost

- After clicking on "+ Category/Subcategory" button below popup will display.
- There are some Value and Name fields. So, Value field will come from Cloud_Computeinventory_#EngagementID# and Cloud_invSubCategoryMapping tables.

Step 1: Select ResourceType as MeterCategory, MeterSubCategory from Cloud_ComputeInventory_#EngagementID#

- Here ResourceType is Metercategory which value will bind in Category value dropdown and second MeterSubCategory which value will bind in SubCategory Value dropdown.

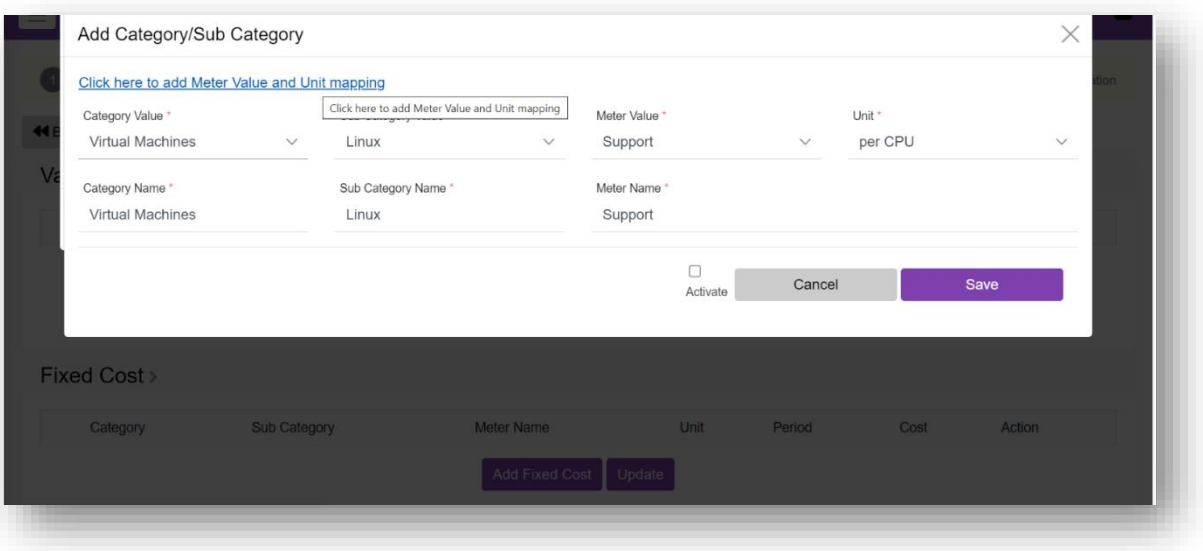

Figure 74 – Cost Model (Cont.)

Here, Metervalue is used in the future classified to Category and subcategory into third subsequent.

E.g:

- Category SubCateogry MeterValue Unit calculated field
- Compute Virtual Machine Support perCPU InventoryVCpu
- Computer Virtual Machine Maintenance perCPU InventoryVCpu

- Name fields are the fields where users can create user friendly names accordingly.
- Users can create and edit categories from separate page also once it's been activated then it can be used.
- Once we choose Category Value, Subcategory value, Meter Value, Unit and user-friendly name (Category Name, Subcategory name and Meter Name)
- Meter Name we can create as dynamic e.g.: CPU #ResourcePlatform#
- While billed will be generated it will convert e.g.: CPU Windows 2016 Datacenter
- If you want to create dynamic Meter Name, then dynamic column name you can encapsulate inside ## and that column name should be in Cloud_ComputeInventory_#Engagement# table if user will use wrong column, it will validation and give a warning message to user. The below function is used to parse MeterName.

```
Select [dbo].[ParseMeterName]('CPU #ResourcePlatform#')
```

- Once you click on the save button to save category it will be inserted into this table(object) Select * from Cloud_CostModelCategoryMaster in this table category will be save Cloud provider, Region and Service Category Type wise. Data will be customer specific.
- After saving data, the popup will close automatically, and that data will populate into first popup (Variable Cost Popup). Now we can select category, subcategory, Meter Name and unit from there and will click on the add button record will be add in Variable Cost section as a temporary. To save this data in backend we need to fill in the cost and click on save or update button. This data will be saved in the object below. In this object data will be Cost model wise and version wise. Once data is saved in the database, it will be changed otherwise it will be white only.
- Select * from CostDriverMapping

- If your cost is in pending state (not activated), you can edit update and remove. So, if you want to remove any line item from Variable cost section then you can simply click on the remove icon it will delete data permanently from backend.

Section 2.2 - Fixed Cost

In this section the user can create a fixed category which will not be related to any inventory data. Saving process is just like Variable in fixed also only difference is that there is no need for inventory data and mapping data.

Section 2.1. I - Add New Category in Fixed Cost

1. To create new fixed category just click on Fixed Cost button -> One popup will open -> click "+ Category/Subcategory" button from this popup to add new category you will get another popup as below mentioned.

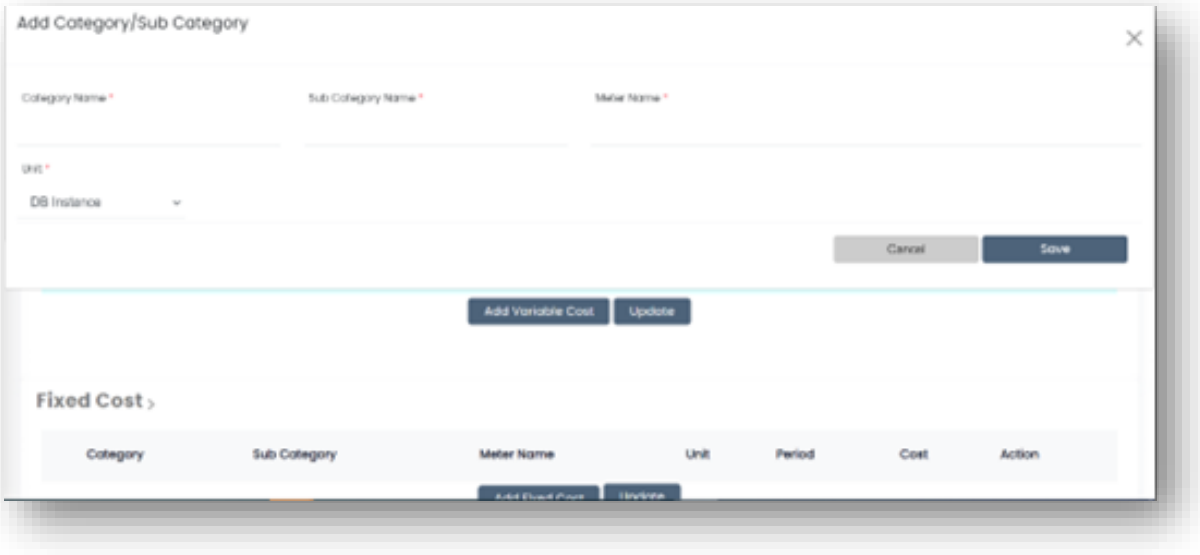

Figure 75 – Cost Model (Cont.)

2. In Above screenshot, you can see there are Name fields only which is user friendly names. Users will enter some names in Category Name, Subcategory Name and Meter Name as a text and unit will be bound from parameter master. See unit data in the query below. If you want a new unit for fixed add unit there.
3. Select * from Parameter Master where vcCode='ServiceUnit'
4. Once you save this data it will be saved in same table as we are saving for Variable only ServiceCategoryType will be Variable/Fixed
5. Table will be Select * from Cloud_CostModelCategoryMaster
6. Now this category will be available in fixed Cost popup. You can simply add this it will add temporarily in fixed Cost section then you can act save/update accordingly.

Cost driver section you can skip this if you do not want to create custom billed.

Custom billed Generation Process

There is a wise provider to create custom billed. Job name like Cloud_AzureCustomBilled_DataProcess if you want to check another provider job just replace provider name into this.

- Switch to DMe now:

```
Select * from jobmaster where jobname like
'Cloud_%%CustomBilled_DataProcess'
```

Table 6 – Customer Billed Data

vcJobId	JobName	JobDescription
B10489	Cloud_AzureCustomBilled_DataProcess	Cloud_AzureCustomBilled_DataProcess
B10492	Cloud_AWSCustomBilled_DataProcess	Cloud_AWSCustomBilled_DataProcess
B10536	Cloud_GeneralCustomBilled_DataProcess	Cloud_GeneralCustomBilled_DataProcess
B10538	Cloud_GCPCustomBilled_DataProcess	Cloud_GCPCustomBilled_DataProcess

- In above template these job ids we are using to create Custom billed generation
- These job Id are of development environment (*******\standard2019)
- Note: sometimes job ID can be different in the release environment so you can find out these by job Name. Job Name will be same
- We have created SQL to SQL Type.
- Exec uspUpdateMultiProviderCostManagementBilledUsageData_Custom @dtFrom = '#StartUpdateDateTime', @dtTo = '#endUpdateDateTime', @EngagementId = '****' , @JobId ='*****' , @ServiceId ='***' , @ToolId ='*****',@CloudProvider='General'
- This procedure will generate Custom Billed

Section 3 - Shared Distribution Section

In this section we can distribute costs. Shared distribution can be done only when we have resourcetag inventory in table Cloud_ResourceTagString_#EngagementID#(live table)

To create shared distribution, we will use the live table and while we distribute cost for a specific month then we will consider the monthly table.

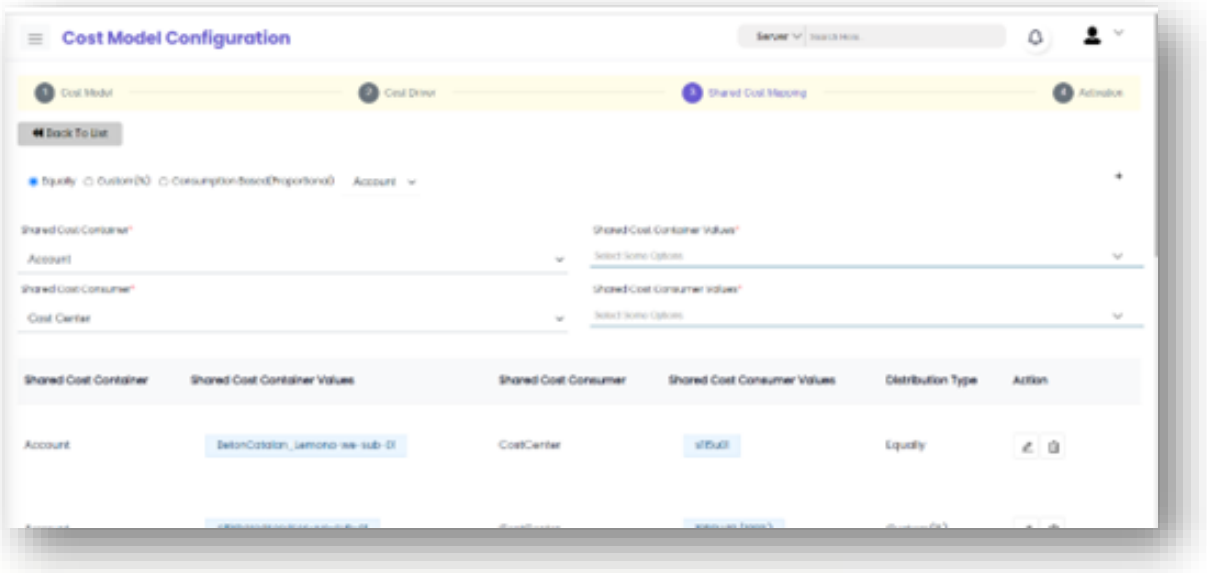

Figure 76 – Cost Model (Cont.)

There are 3 types of distribution:

1. Equal

2. Custom (%)
3. Consumption Based (Proportional)

- **Shared Cost Container** – this is dimension through which we will decide that is the area which we need to distribute. E.g.- This is a field which we want to distribute anywhere.

Right now, we are using Account (Subscription)/Resource Group/Cost Center as a shared container. If you want to add another one yes you can add new, but that column should exist in Cloud_ResourceTagString_#Engagementid# inventory.

- **Shared Cost Consumer** – consumer in which we will distribute cost.

If you want to add some value in Shared Cost Container and Shared Cost Consumer, then insert value in CostModelDimensionMaster table. In this table, there is a dim level field so for Container make it 1 and for Shared Cost Consumer make it 2.

Table 7 – Shared Distribution

Column_name	Description
Id	Auto incement field
LabelName	User-friendly Name
DimensionValueId	This value will be bind in dropdown Value that should be matched with Cloud_ResourceTagString_#EngagementID# columns
DimensionTextId	This field is used for dropdown text
KPIId	This is not used right now
FilterParentId	This will be mapped with Shared Cost Container value in dim level 2 only
DimLevel	dim level 1 means Shared Cost Container value 2 means Shared Cost Consumer value
EngagementId	Customerid mandatory column
ToolId	This is tool id mandatory column
Description	to save description but right now no reference of this column
IsActive	If Y then the record will display else not
CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
bCustom	

- The below Store procedure is used to bind Shared Cost Container dropdown here dim Level will be 1

```
exec GetCostModelDimension
@DimensionId=NULL,@engagmentId=N'BC01',@DimLevel=1,@FilterParen
tId=NULL,@ToolId=N'****'
```

- The below Store procedure is used to bind Shared Cost Consumer dropdown here dim Level will be 2 and FilterParentId will be the value of Shared Cost Container value

```
exec GetCostModelDimension
@DimensionId=NULL,@engagmentId=N'BC01',@DimLevel=2,@FilterParen
tId=N'subscription',@ToolId=N'****'
```

- The Engagement ID we have kept on the table but is not used because this data will be common for all customers and will be as master data.
- Come to the Shared Cost Container Values this data will fetch from -
  Select * from Cloud_ResourceTagString_#EngagementID#
- This is Cloud_ResourceTagString_#EngagementID# table definition:

Table 8 – Cloud_ResourceTagString

Column_name	Column_Descrition
Customer	Customer Name
CustomerId	Customer Id
ChildAccountId	Subscription (Account) ID
ResourceId	Resource Id
resourcename	Resource Name
tags	Tag data in json format
Active	Active flag
createddate	
Environment	this is environment tag value which will comes from originalTag tag column mapping you can check in TagMapping table
Region	Resource Region
Location	Resource Location
Department	this is Department tag value which will comes from originalTag tag column mapping you can check in TagMapping table
Organisation	This is Organization tag value which will come from originalTag tag column mapping you can check in TagMapping table
Application	This is Application tag value which will comes from originalTag tag column mapping you can check in TagMapping table
OriginalTag	This is tag data in json format
BusinessUnit	This is BusinessUnit tag value which will come from originalTag tag column mapping you can check in TagMapping table
CostCenter	This is Application tag value which will comes from originalTag tag column mapping you can check in TagMapping table
Owner	This is Owner tag value which comes from originalTag tag column mapping you can check in TagMapping table
Project	This is Project tag value which will come from originalTag tag column mapping you can check in TagMapping table
Column1	this we can use for any new tag
Column2	this we can use for any new tag
Column3	this we can use for any new tag
Column4	this we can use for any new tag
Column5	this we can use for any new tag
Parent	If Resource have parent, then you will find data here
cloudprovider	Provider Name
UsageMonth	Resource inventory month date

Table 9 – Cost Model Values

Column_name	Column_Descrition
nmAutoId	auto incement column
Field	This is the same name which tag column is used in Cloud_ResourceTagString_#EngagementID# table
TagHeaderValue	This Original Tag key value
Provider	provider name
CustomerName	pass Engagement id here if you want to define tag customerwise
vcCreatedBy	
dtCreatedOn	
vcModifiedBy	
dtModifiedOn	
TagSequence	This sequence will be same as it is because in this seq., we are inserting data in tag columns of Cloud_ResourceTagString_#EngagementID# table

```
exec GetCostModelDimensionValues
@DimLevel=2,@engagmentId=N'****',@DimensionId=N'CostCenter',@Fi
lterDimensionId=N'subscription',@FilterValue=N'|BetonCatalan_Le
mona-we-sub-01',@CloudProvider=N'AZURE',@ToolId=N'****'
```

- Shared Cost Container values will be bound from Cloud_ResourceTagString_#Engagement# table on basis of selected SharedCostContainer.
- Shared Cost Consumer Values will be bound by Shared Cost Container value and Selected Shared Cost Consumer. There is a dropdown nearby by distribution Type here will be Overall and selected value of Shared Cost Container so if we choose Overall then Container will be all add in CostCenterDepartmentMapping_#Engagement# for CRH Customer only.
- So, if we are saving Overall then distribution will be on overall selected Shared Cost Container other distribution will be of specific value e.g.- If we are choosing Subscription in Shared Container and Distribution on Overall then distribution will be on overall subscriptions else distribution share will be on subscription wise.

Shared Matrix Table Details

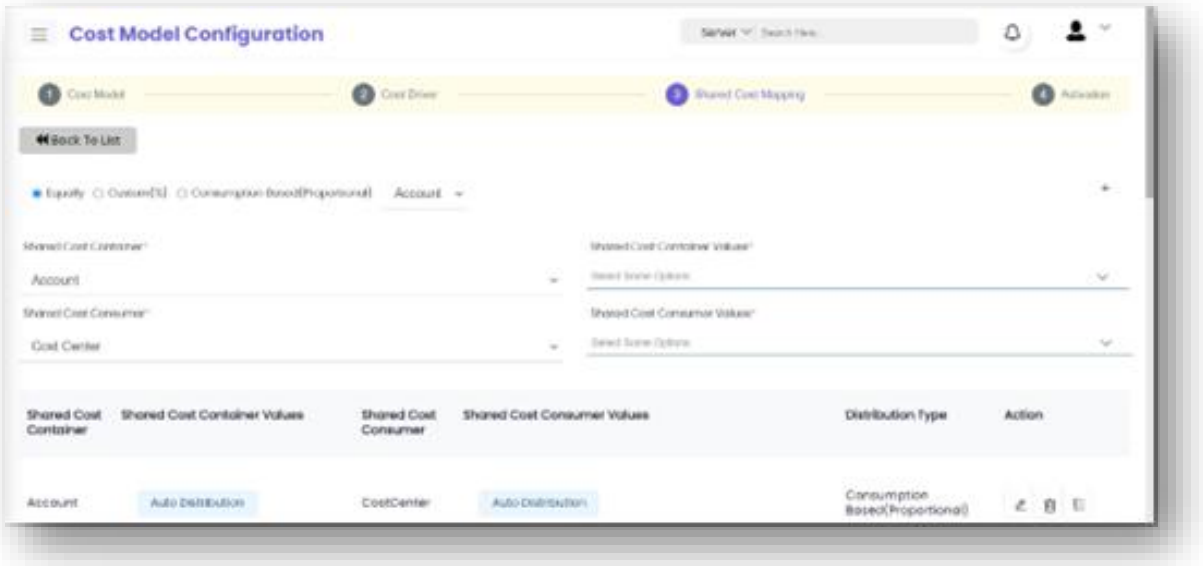

Figure 77 – Cost Model Configuration (Cont.)

Figure 78 – Cost Model Configuration (Cont.)

```
Select * from SharedCostDistributionMaster where
CostModelId='CM1'
```

```
Select * from SharedCostDistributionDetail where
SharedCostMatrixId in (Select ID from
SharedCostDistributionMaster where CostModelId='CM1')
```

The above two tables will be used for shared Matrix.

Table 10 – SharedCostDistributionMaster

Column_name	Description
Id	unique id of table

CostModelId	Cost model id
CostModelVersion	Cost model version
ToolId	tool id will be nmServiceSubTypeID
DataSet	not in use but saving same as in dimensionId
DimensionId	Shared Cost Container dropdown value
ChildDimensionId	Shared Cost Consumer dropdown value
DistributionType	Distribution Type like equal/Custom/Consumption
DistributionValue	Pipe separated value of Shared Cost Container
DistributionUnit	Distribution Unit
bCustom	not in use but future reference
IsActive	
CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
DistributionValueIds	not in use
EngagementId	Customer id
DistributionOnProportional	Overall/selected value of Shared Cost Container
IsAutoDistribution	In Consumption we can do auto distribution

Table 11 – SharedCostDistributionDetail

Column_name	Description
Id	unique id of table
SharedCostMatrixId	Unique id of SharedCostDistributionMaster
ChildDimensionId	Shared Cost Container dropdown value
ChildDimensionValue	Shared Cost Container Values dropdown value
DistributionValueShare	Percentage share in case of custom
IsActive	
CreatedBy	
CreatedOn	
ModifiedBy	
ModifiedOn	
ToolId	
bCustom	
Distribution	Right now, we are not using this, but value is same as in DistributionValueShare

Select * from Cloud_SharedDistribution_Log

- We Can edit shared matrix items before activation or also can delete.

- We can view Consumption distribution percentage shares on click of view icon in corresponding Shared matrix items.

Consumption percentage we are calculating internally in procedure, but we are storing in Cloud_SharedDistribution_Log table.

Equal Distribution:

- In equal distribution, distribution share will be calculated by (1/count of dimension) *100
- The number of dimensions will be counted from SharedCostDistributionDetail

  e.g.- if there are 2 Containers then (½*100%) =50. 50-50%

  First, we find out Total shared Cost On the basis of Shared Cost Consumers then will distribute 50-50% to the matrix Shared Cost Containers.

Custom Distribution:

- In Custom distribution, distribution share we can save while creating Custom shared matrix.
- First, we find out Total shared Cost On basis of Shared Cost Consumers then will distribute Shared Distribution values to the matrix Shared Cost Containers.

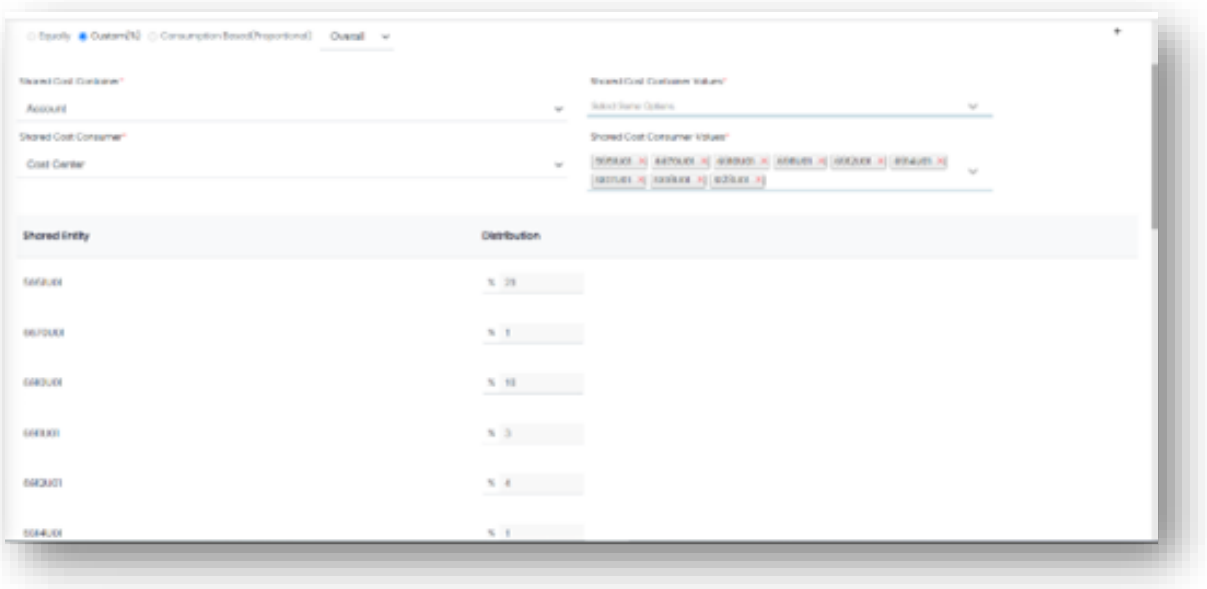

Figure 79 – Cost Model Configuration (Cont.)

Consumption Based (Proportional) Distribution:

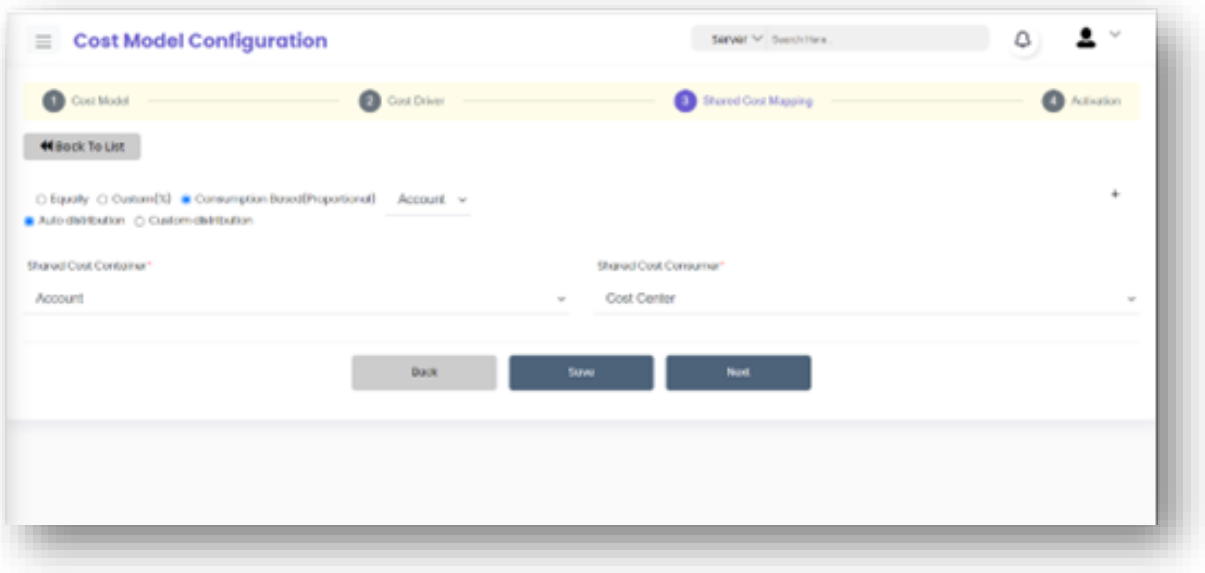

Figure 80 – Cost Model Configuration (Cont.)

- In Consumption distribution, we can create 2 types of matrixes:
  - Auto distribution
  - Custom distribution
- In Auto Distribution will be automatically on all Shared Cost Container values except to use in other matrix. So, once we choose the Auto distribution option for the values options will not need to choose. Values option will be hidden in the case of Auto Distribution.
- In Custom distribution, we will choose Values also.
- The percentage will be calculated Total cost Shared Cost Container wise / Sum (Container Cost)
- First, we find out Total shared Cost then will distribute Shared Distribution values to the matrix Shared Cost Containers.

DME Job Process:

First, we collect all Billed/Unbilled data for specific providers Azure/AWS/GCP in below destination table Customer wise and billed will be Year and month wise.

Below mention table are the dedicated table for native data.

- Azure Billed/Unbilled destination tables

```
Select * from CloudData_AzureCSP_Billed_#EngagementID#_2022_08

Select * from CloudData_AzureCSP_unBilled_#EngagementID#
```

- AWS Billed/Unbilled destination table

```
Select * from AWS_S3_Billed_#EngagementID#_2022_08

Select * from AWS_S3_unBilled_#EngagementID#
```

- GCP Billed/Unbilled destination table

```
Select * from GCP_Billed_#EngagementID#_2022_08
```

If there is any custom Billing, then run those job Provider wise as mentioned below:

Table 12 - Provider Details

B10479	VCenter_CostManagement_Billed_DataProcess
B10489	Cloud_AzureCustomBilled_DataProcess
B10492	Cloud_AWSCustomBilled_DataProcess
B10536	Cloud_GeneralCustomBilled_DataProcess
B10538	Cloud_GCPCustomBilled_DataProcess

- This data will directly push into the final table Cloud_SharedCost_Billed_#EngagementID#_2022_08
- This process is an option. Users only need to do this while he/she creates some custom line item in Cos driver in Cost model.
- Next, above all native tables data we will merge into the final table after doing discount.
- So, after doing discount we are storing discount amount in pricingpretaxTotal and pricetocusomterAmount.
- So, if shared matrix will be saved in cost model, then shared logic will be applied in the final table.
- So, the final table structure is just like Azure CSP table. A few columns are added newly which is mentioned below:

Table 13 - Cloud_SharedCost_Billed_#EngagementID#

Column Name	Status	Remarks
SavingsForCustomerUSD	Old column	Storing original value of tobechargequantity
OriginalCost	New column	Storing original value of pricingtopretaxtotal
RecordType	New column	This will status like original/Updated/Shared
ServiceCategoryType	New column	this will come from cost model bill data only it will null in native. Value will be like Variable/Fixed
IsSharedCost	New column	This is flag which will be 1 as per the shared rule
CostCenter	New column	This is Container value
AdditionDetailsJson	New column	In this column we can save Json.e.g. CRH is using custom billing so they want extra column that we can save in Json and will parse those in metric store procedure
PriceToCustomerAmount	Old column	Discounted value
PricingPreTaxTotal	Old column	Discounted value

- This is the final table Select * from Cloud_SharedCost_Billed_#EngagementID#_2022_08
- After doing the above process we will run process jobs.
- Once will run daily just like unbilled job and second one will run after billed data will be collected from native just like Billed job so will save this job time after billed jobs.
  e.g.- billed jobs will run on 9th of every month to collect billed data from previous month then we save 10th or 11th date in B10537 job.

B10490	Cloud_SharedDistribution_DataProcess
B10537	Cloud_SharedDistribution_DataProcess_Monthly

- Now we will check the data in the final table. If data is found, then we can switch to matrix.

Metric Collection Configuration:

- Below mentions metric, we are using for multi cloud billing.
- So will save new metric store procedure as in screenshot below and will add new added column in metric columns.
- Previously we were using Cloud_ExchangeRatesAzuretable for exchange rate but now we change this to Cloud_CurrencyExchangeRate. We are using the same.
- Currency exchange job B9302 Cloud_ExchangeRateAzure

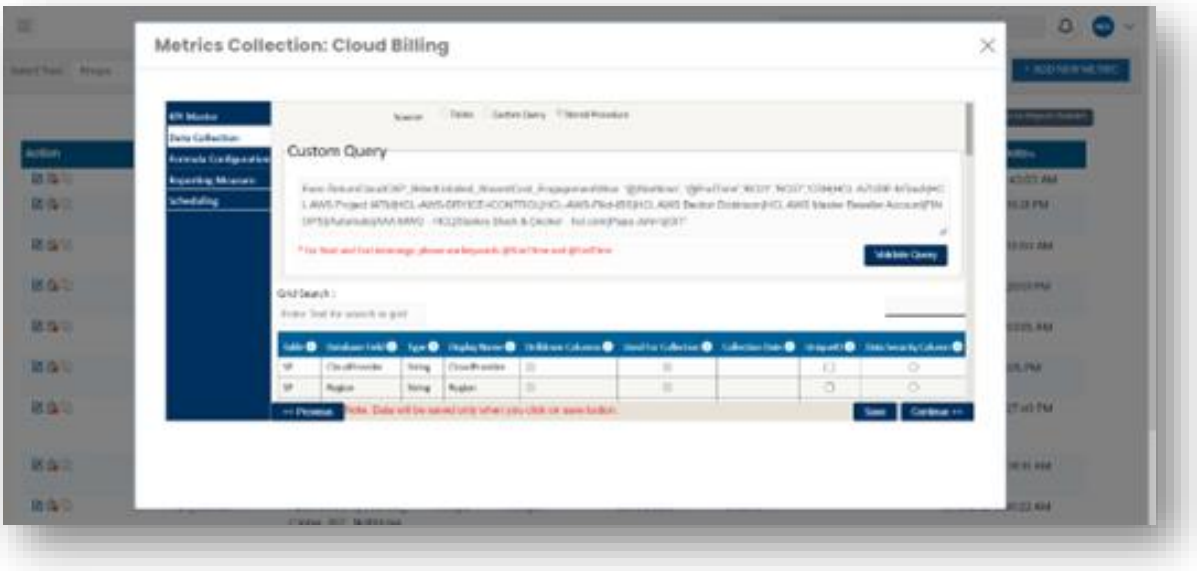

Figure 81 – Cost Model Configuration (Metric Configuration)

- We will save reconciliation for this metric so that metric can run automatically after process job will run for previous month so will keep this date after then monthly process job. Here it is 14th date.

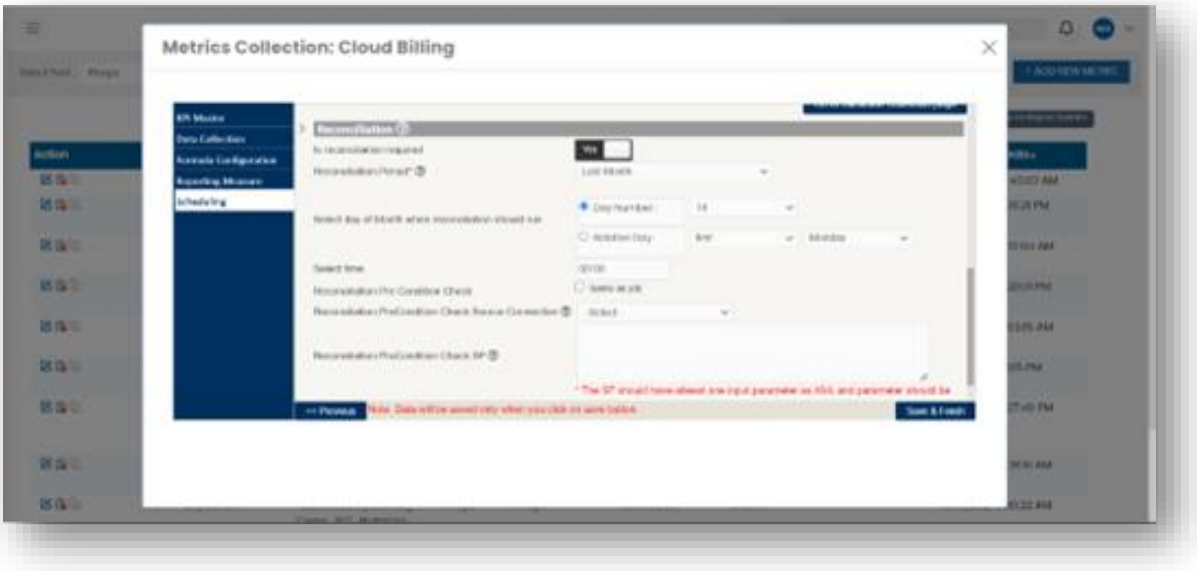

Figure 82 – Cost Model Configuration (Metric Configuration) Cont.

4. We will save reconciliation for this metric so that metric can run automatically after process job will run for previous month so will keep this date after then monthly process job. Right now, I am using 14th.

Scheduled Metric Collection:

- After doing metric collection configurations we can schedule metric for multi cloud Billing.

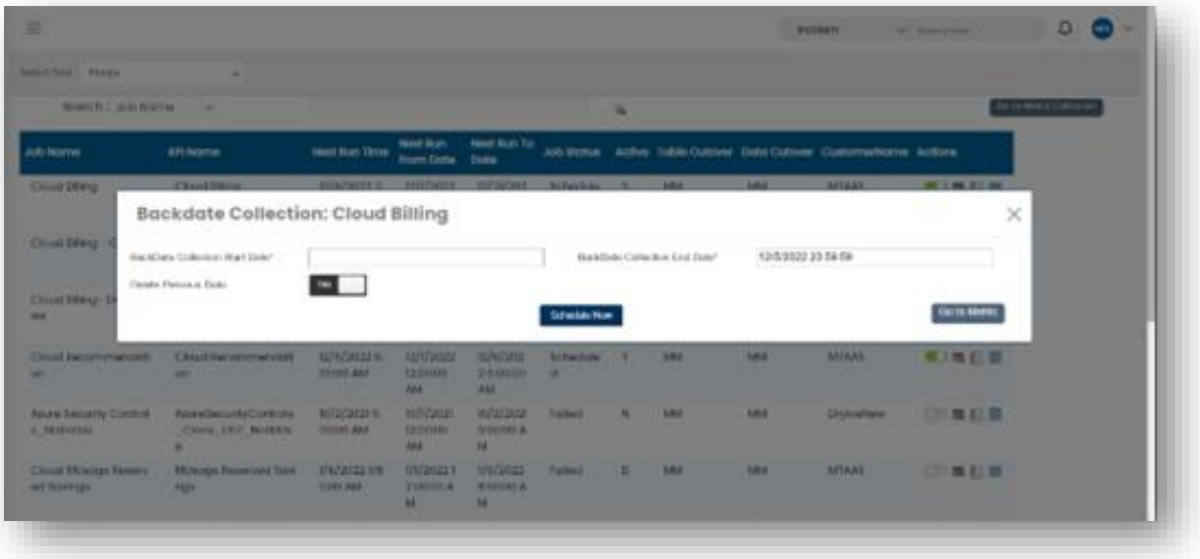

Figure 83 – Cost Model Configuration (Metric Job Schedule)

Universe Configuration:

- Now we will configure universe configurations. First will add newly added columns.

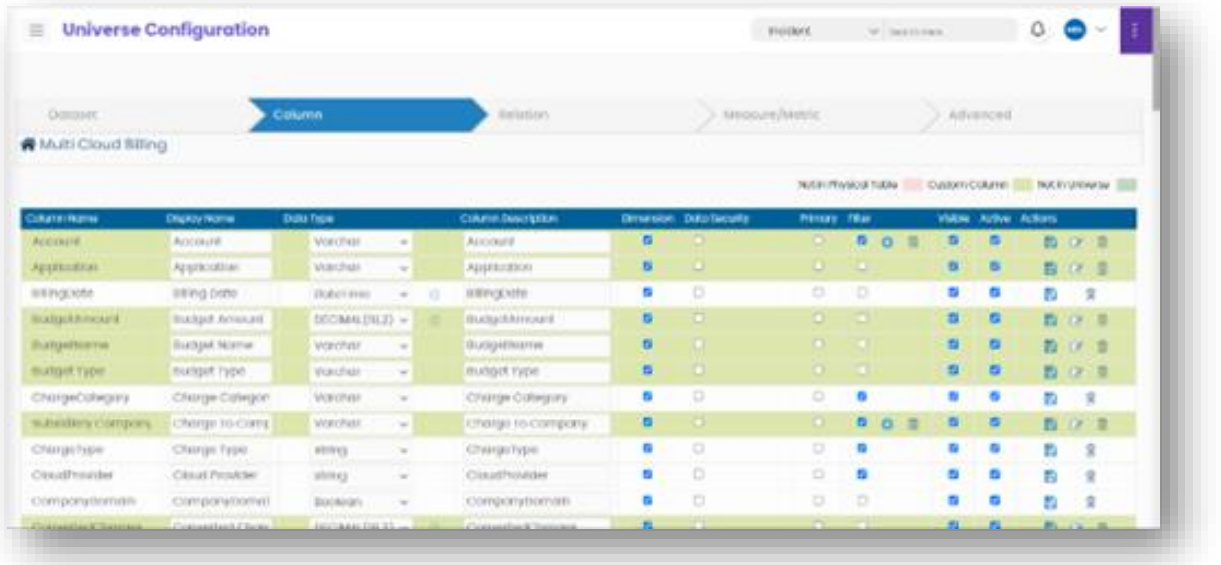

Figure 84 – Cost Model Configuration (Universe Configuration)

- We added budget and Currency conversion joins here and done the changes in those columns where we were using these joins.

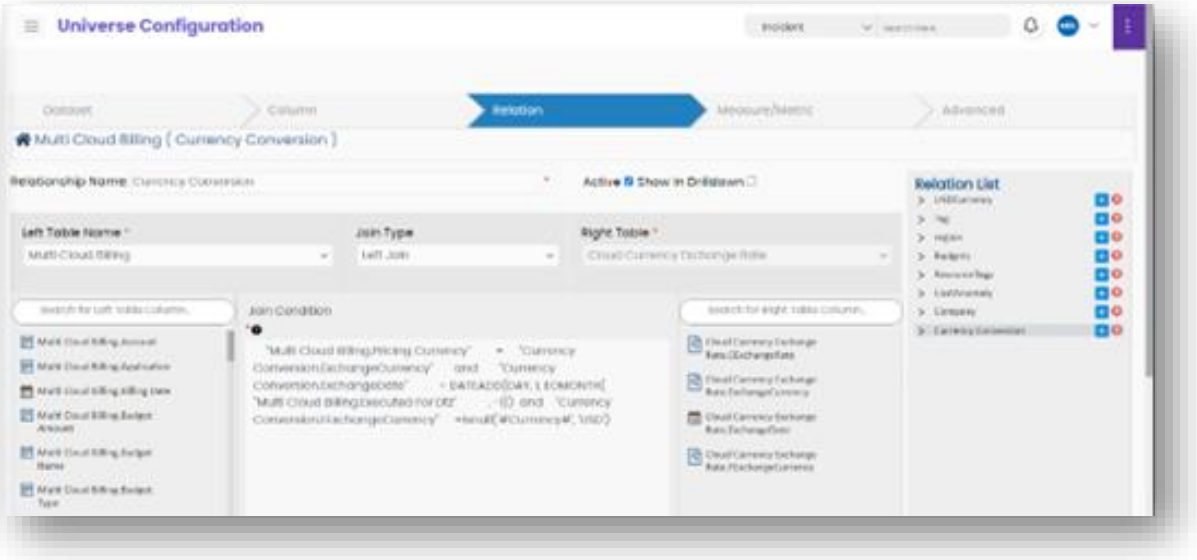

Figure 85 – Universe Configuration (Cont.)

To navigate to Shared Cost Rule Configuration, go to Menu -> Configuration -> FinOps -> Shared Cost Rule.

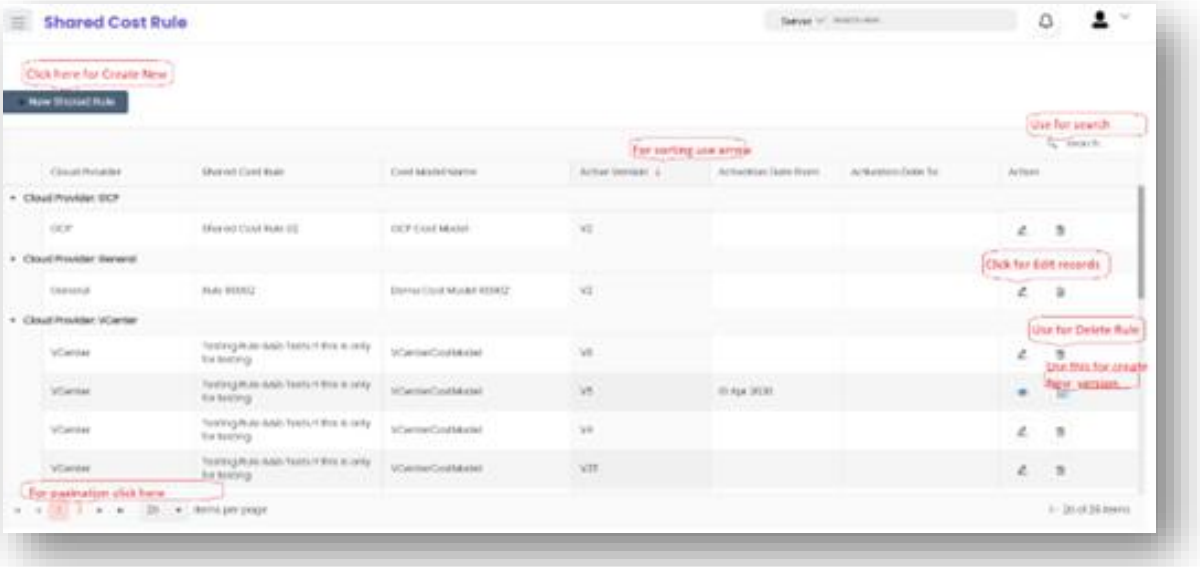

Figure 86 – Shared Cost Rule

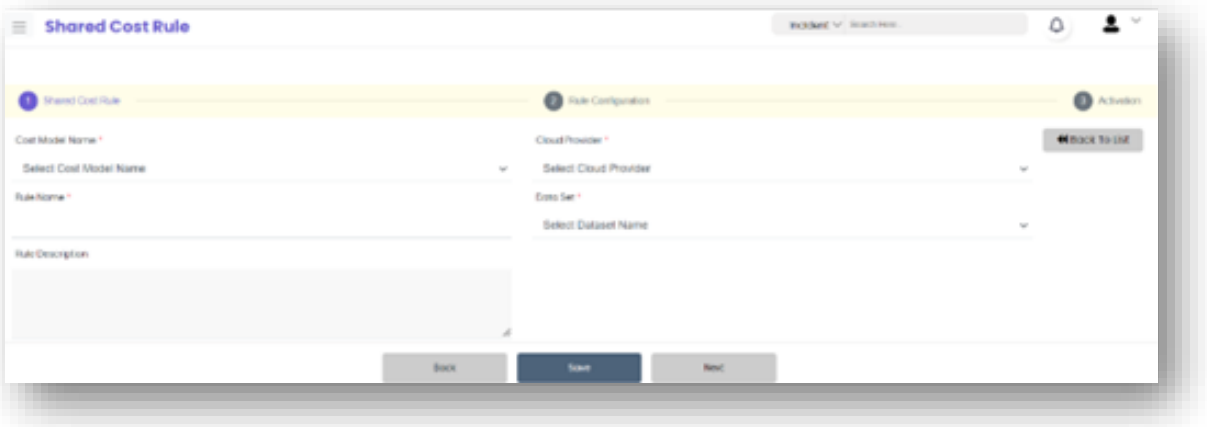

Figure 87 – Shared Cost Rule (Cont.)

- Cost model Name- Cost model name comes from cost model screen which should activated with version
- Using Stored proc exec GetCostModelDetail @customerid='****',@toolid='****' we are getting cost model name version should active and cost model should activate.
- Cloud Provider dropdown is to select the provider's name. The method is already defined in the CostModel page. We will be changing the backend query of this SP to populate the dropdown at a later stage.
- Cloud Provider is binding from the object below:

```
SELECT vcvalue as ID,  vctext as [Name] from ParameterMaster
where  vcCode =     'CloudProvider'
```

```
Procedure - exec GetCostModelCloudProvider
cloudProvider=NULL,@engagmentId=N'#EngagementID#',@toolId=n
'***'
```

- Rule name is for giving a unique name to the rule
- The rule description is to give some details for the Rule.
- Dataset: Dataset is shown here as the rules will be based on the fields of this table and values in them. Dataset value in dropdown Cloud_SharedCost_Billed
- By using stored proc uspAddSharedCostRule rule is creating.

Rule Configuration Step 2:

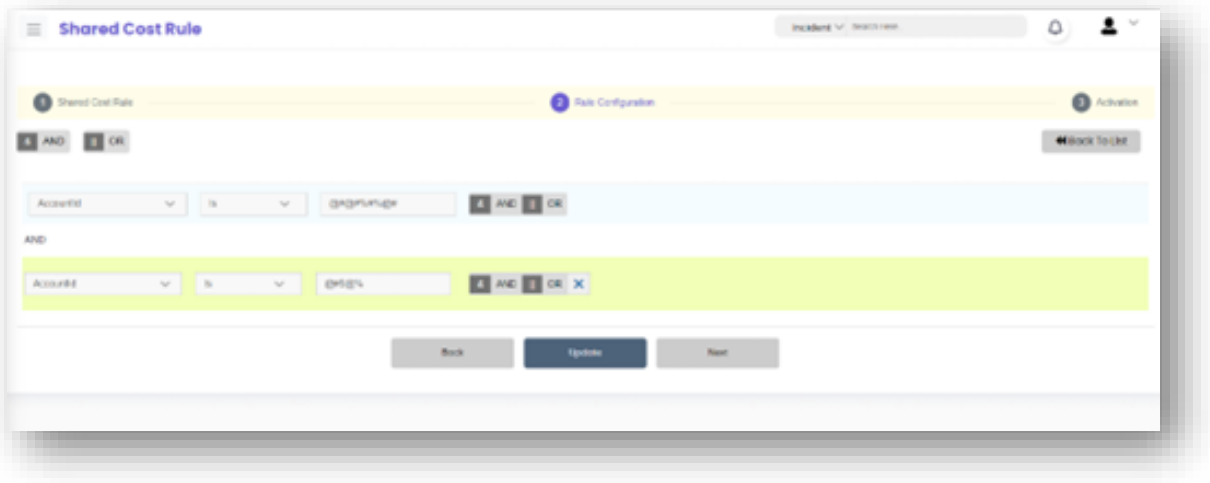

Figure 88 – Shared Cost Rule (Cont.)

- This screen is allowing user to create the actual rule based on the fields available under the table "Cloud_SharedCost_Billed_#EngagementID#" this value in yellow is dynamic. Based on the user's engagement this table is dynamically picked.
- Column bind from table "Cloud_SharedCost_Billed_#EngagementID#" using stored procedure.

```
exec [uspGetSharedCostColumns]
@EngagementId=N'#EngagementID#',@toolId=N'***',@FilterCode=N'Co
mpany'
```

Table 14 – SharedCostRuleMaster

Column_Name	Type
Id	varchar
SharedCostRuleName	varchar
nmServiceSubTypeID	varchar
vcEngagementID	varchar
CloudProviderId	varchar
vcProfileId	varchar
SharedCostRuleNameDescription	varchar
IsActive	char
CreatedBy	nvarchar
CreatedOn	datetime
ModifiedBy	nvarchar
ModifiedOn	datetime
StartDate	datetime
bCustom	bit

Table 15 – SharedCostRuleDetails

Column_Name	Type
Id	int
SharedCostRuleId	nvarchar
ShareCostRuleVersionNo	int
FilterJSON	varchar
FilterXML	varchar
IsActive	char
CreatedBy	nvarchar
CreatedOn	datetime
ModifiedBy	nvarchar
ModifiedOn	datetime
nmServiceSubTypeID	varchar
vcEngagementId	varchar
bCustom	bit

The important field in this table is FilterJSON and FilterXML. The conditions which user will create will be saved in these 2 fields in 2 formats.

Sample conditions created are shown below followed by the resulting XML which needs to be saved.

Sample XML format is shown below:

```
<FilterCondition>
    <cAdvFilterRow>
      <rowid>1</rowid>
      <rowparentid>-1</rowparentid>
      <rowlogicaloperator>AND</rowlogicaloperator>
      <rowfieldvalue>BC27243</rowfieldvalue>
      <rowfieldtext>Meter Category</rowfieldtext>
      <rowfielddatatype>datatype_textRefer</rowfielddatatype>
      <rowoperatorvalue>Is Any Of</rowoperatorvalue>
      <rowoperatortext>Is Any Of</rowoperatortext>
      <rowselectedvalue>Virtual Machine|†|Virtual Machine
Licenses|†|Virtual Machines</rowselectedvalue>
      <rowfilterorder>0</rowfilterorder>
    </cAdvFilterRow>
    <cAdvFilterRow>
      <rowid>2</rowid>
      <rowparentid>1</rowparentid>
      <rowlogicaloperator>AND</rowlogicaloperator>
      <rowfieldvalue>BC27255</rowfieldvalue>
      <rowfieldtext>Region</rowfieldtext>
      <rowfielddatatype>datatype_textRefer</rowfielddatatype>
```

```
      <rowoperatorvalue>Is Any Of</rowoperatorvalue>
      <rowoperatortext>Is Any Of</rowoperatortext>
      <rowselectedvalue>Switzerland</rowselectedvalue>
      <rowfilterorder>0</rowfilterorder>
    </cAdvFilterRow>
    <cAdvFilterRow>
      <rowid>3</rowid>
      <rowparentid>-1</rowparentid>
      <rowlogicaloperator>OR</rowlogicaloperator>
      <rowfieldvalue>BC27261</rowfieldvalue>
      <rowfieldtext>Subscription</rowfieldtext>
      <rowfielddatatype>datatype_textRefer</rowfielddatatype>
      <rowoperatorvalue>Contains</rowoperatorvalue>
      <rowoperatortext>Contains</rowoperatortext>
      <rowselectedvalue>Shared</rowselectedvalue>
      <rowfilterorder>0</rowfilterorder>
    </cAdvFilterRow>
  </FilterCondition>
```

Validations:

- In this phase, we will not allow users to save multiple rules against a single cost model. He can save one rule for different cloud providers (AWS/GCP/Azure/Vcenter).
- Users cannot edit the Shared_Cost_rule if respective cost model is activated.
- A rule cannot be deleted once respective cost model is activated.
- Dataset in the page will be restricted to just one dataset only “Cloud_SharedCost_Billed_#EngagementID#”. If this object for the engagement does not exist, then users should not be allowed to create a rule or proceed further.

FinOps Shared Cost V2.0:

In MyXalytics-FinOps (MtaaS) multitenant console needs to be given to a user (admin role) of a customer to define the rule(conditions) for identifying the data points/lineitems/records which needs to be marked Shared. The rules which will be created from the console will be used in Cost model data processing job to mark the billing line items as Shared.
As such no existing module will be impacted due to this feature but below modules need to be updated to use the rules.
Below are the key enhancements/Changes are required in various FinOps components to achieve this feature.

- A new Tile “Shared Cost Rules“needs to be added to the menu for this console. MYXA admin will add this as it is a one-time job.
- New permission/RBAC needs to be defined for this page and used so that authorized users can access the page.
- Exception logging needs to be done for the page.
- Loging needs to maintain on a separate table in Json form to keep all the details of the record before any changes are saved. Json format is suggested as this just a log of the old record before changes were saved.
- Methods need to be created in the WEB API solution and as Internal type.

UI and Component Changes:

The components mentioned below are the components which will change due to the configuration.

- WEB API solution
- FinOps Database

Web API:

Respective methods have been exposed to the Web API to let the web application access them. These methods are not to be consumed by external applications/tools but only MyXalytics application. These methods will be exposed as an internal end point so that only MyXalytics application can access them using the Service principal App for MyXa.

The methods which are required for this page and are available in old Web App solution need to be brough to the WEB API solution.

5 Event Rule

This module helps in configuring rules for Compliance for generating alerts and sending emails based on defined ranges.

Rules can be defined for:

- Compliance

5.1 Rule Configuration

Steps to configure the rule:

1. Go to Configuration tab -> Finops -> Compliance -> Event Rule:

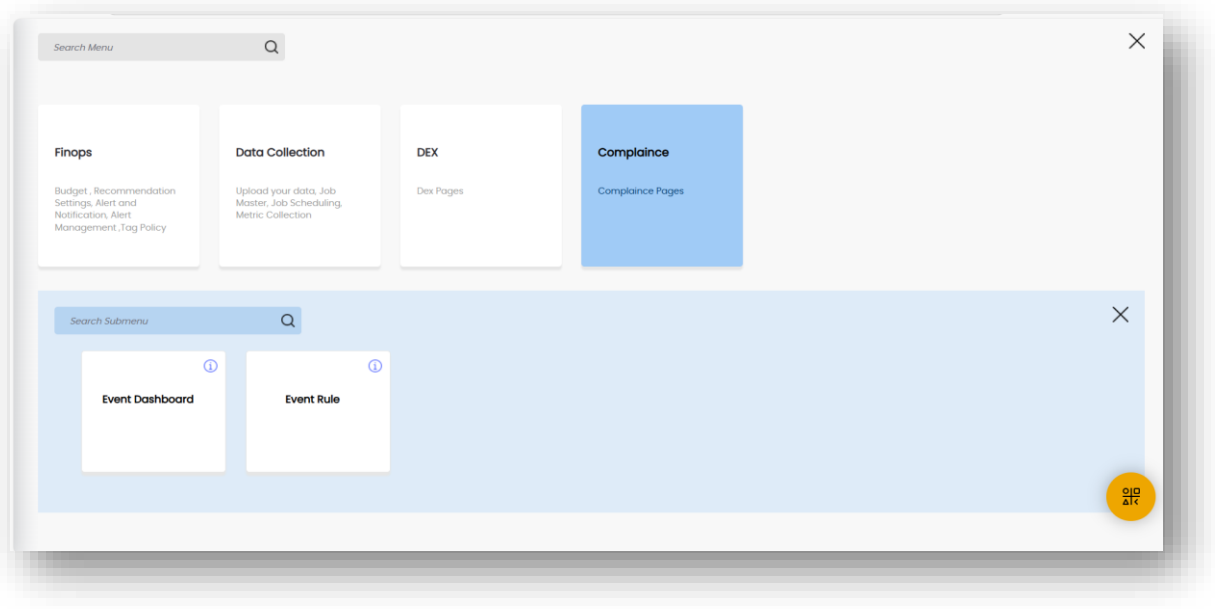

Figure 89 – Event Rule

2. Click on the Create New Rule button on top right.

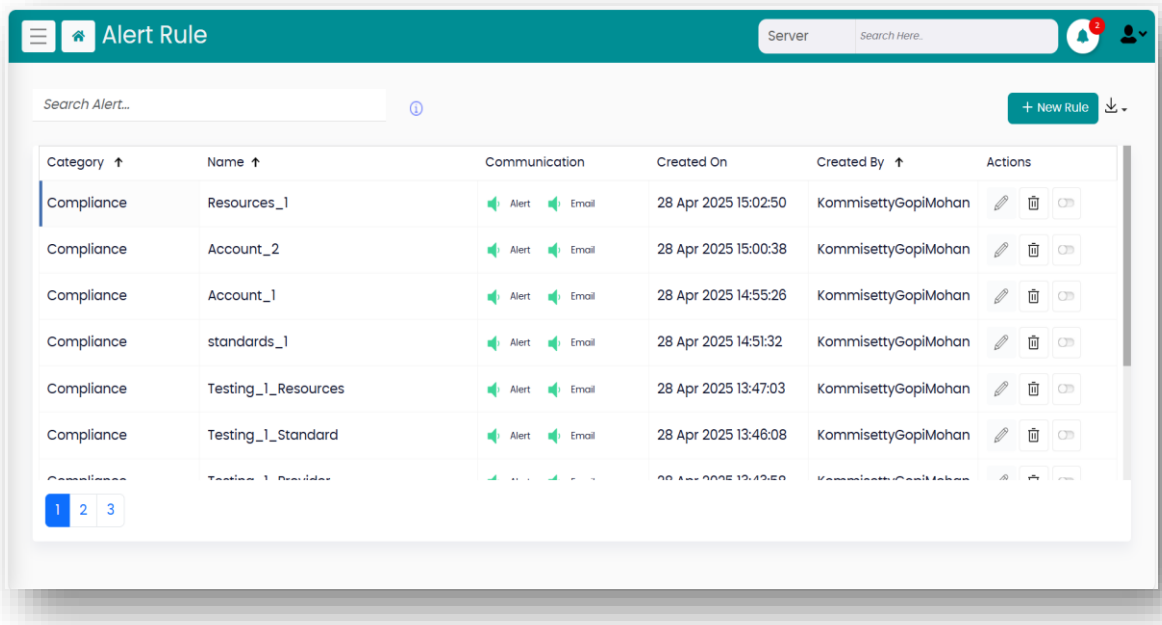

Figure 90 – Event Rule

3. In the below page, following fields should be filled in:
   - **Alert Name**: Enter a name for the alert rule.
   - **Alert Category**: Configuration to be done for which category: Compliance

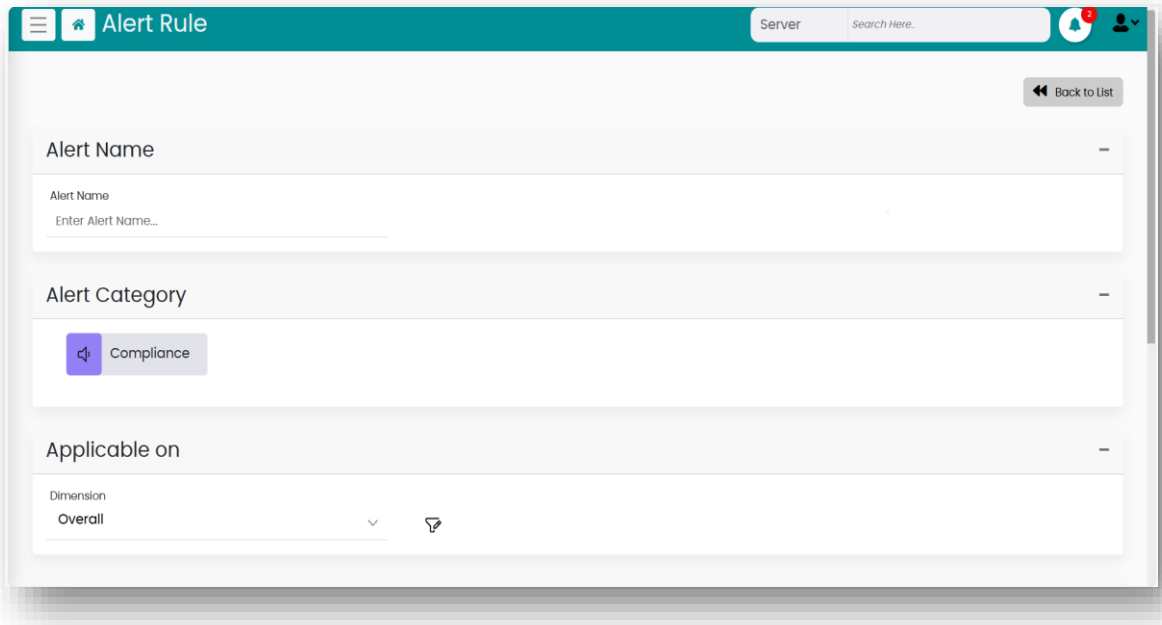

Figure 91 – Event Rule

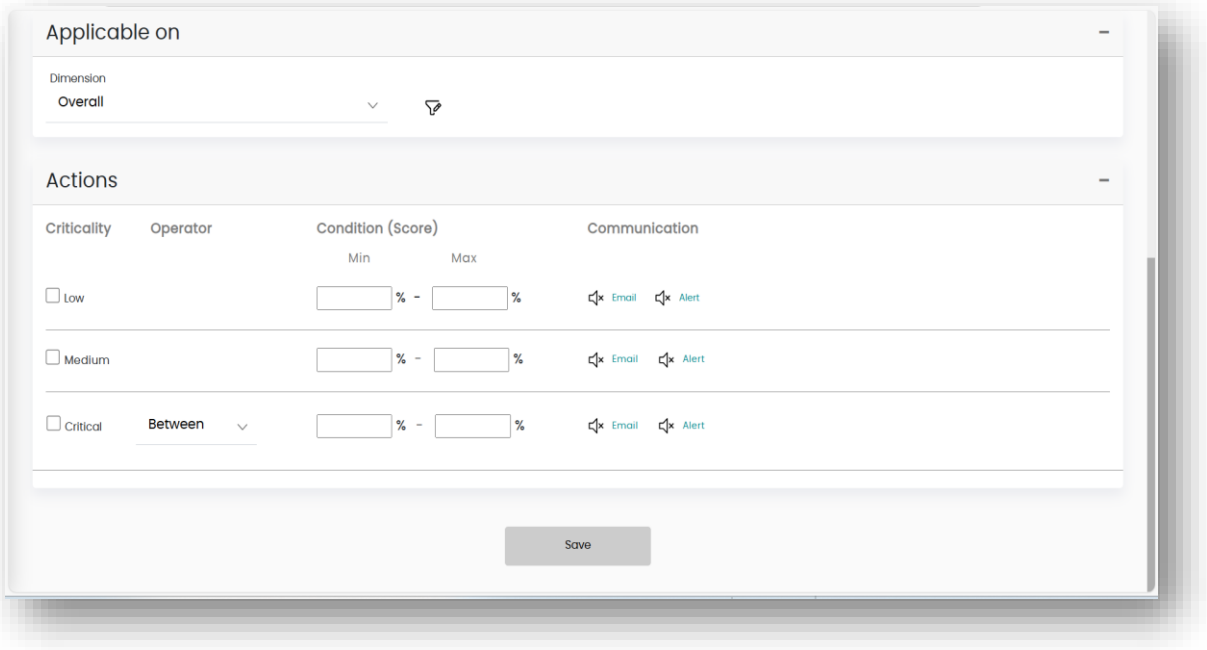

Figure 92 – Event Rule

– **Applicable on**: Dimension for which the rule will be applicable. Options are **Overall, Account**, Provider, Standards, Resources.

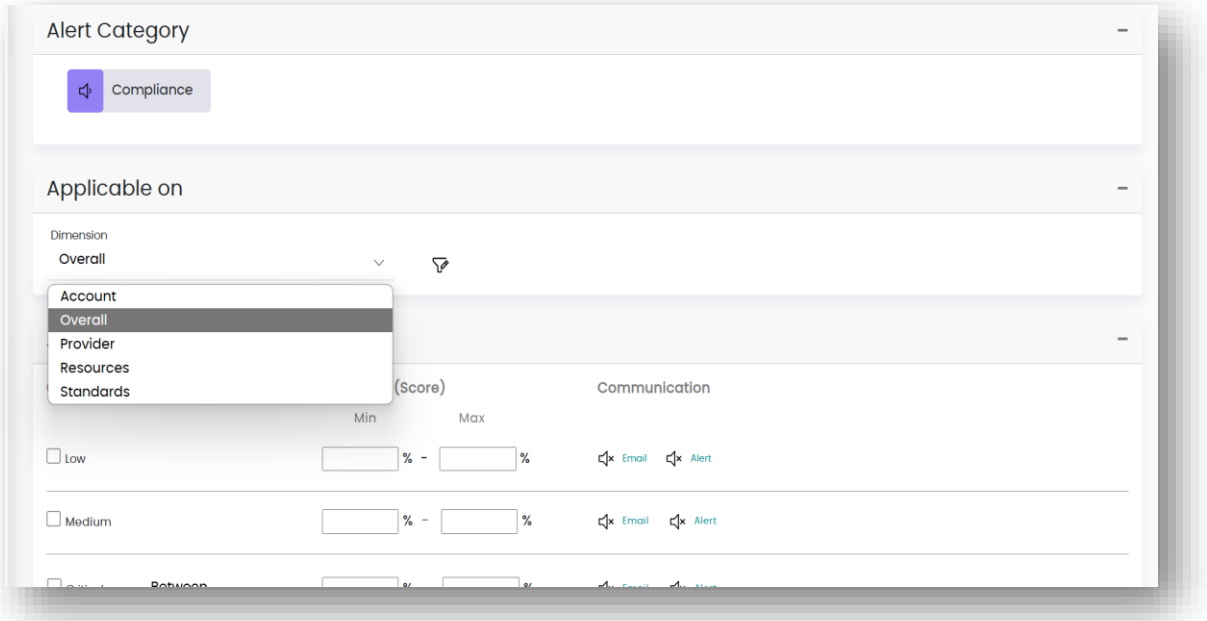

Figure 93 – Alert Parameters

– **Lookback Period**: This is by default: This Month

– **Actions**: User needs to define the ranges and severity for which he wants the email or/and alerts.

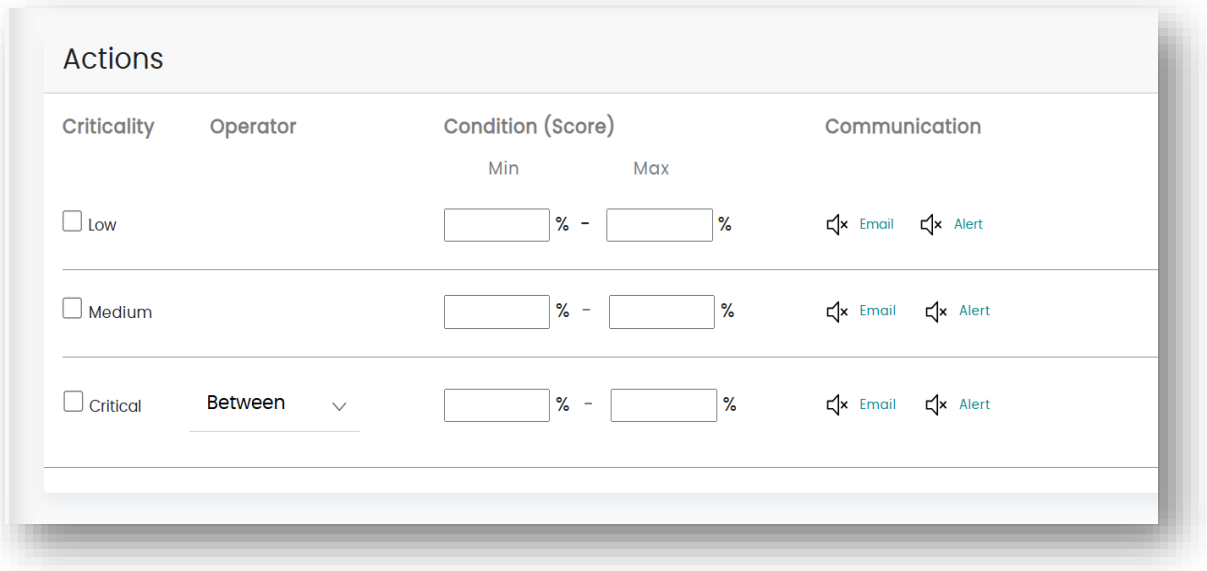

Figure 94 – Alert Actions

The email address to which mail should be sent should be configured by clicking on Email/ Alert links under Actions.

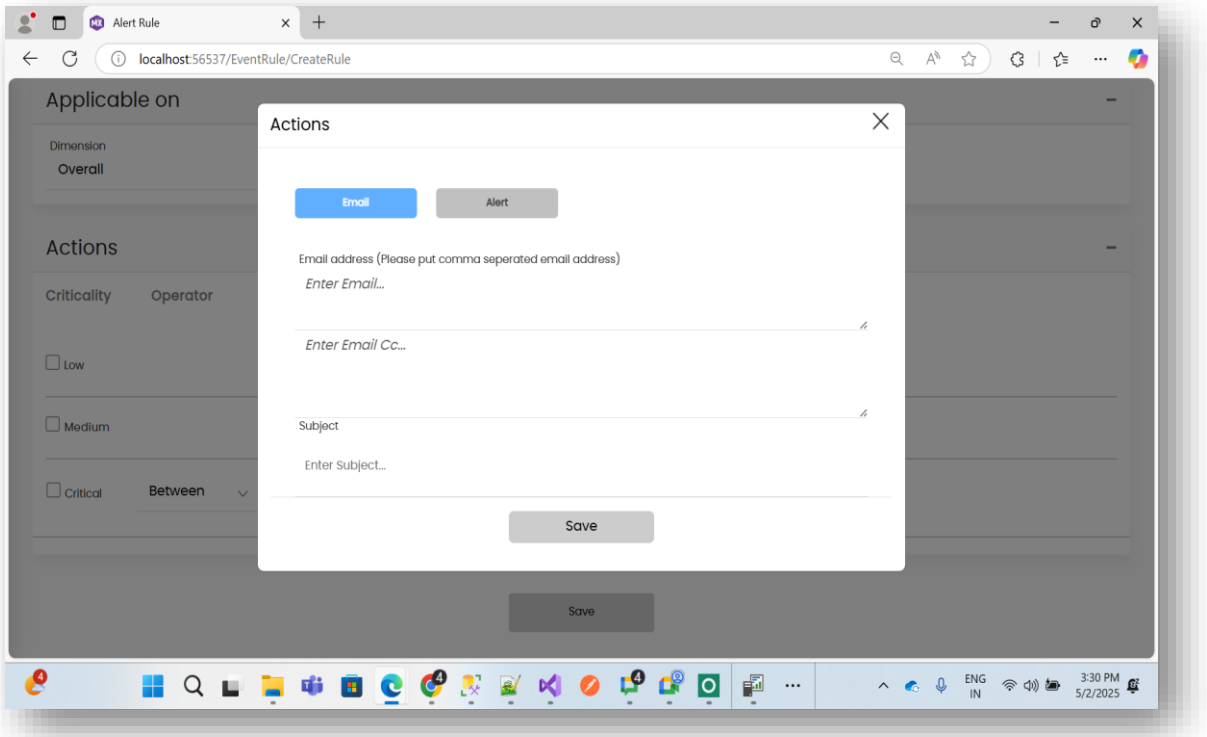

Figure 95 – Alert Rule Actions

4. Click on the Save button to save the configuration.
5. The listing page enables user to Edit, Delete and Enable/ Disable a rule.

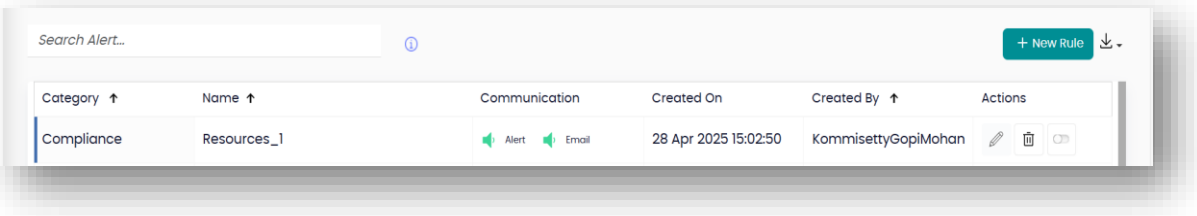

Figure 96 – Edit Alerts

Here we can see different tabs technique wise. Each tab shows savings per resource. These can be assigned to assignees with severity. As soon as a line item is assigned to someone, an alert is generated. The alerts can be seen on Event Dashboards.

5.2 Event Dashboard

Users can go to Event dashboard by following below path:

1. Menu -> Configuration -> Finops ->Compliance -> Event Dashboard

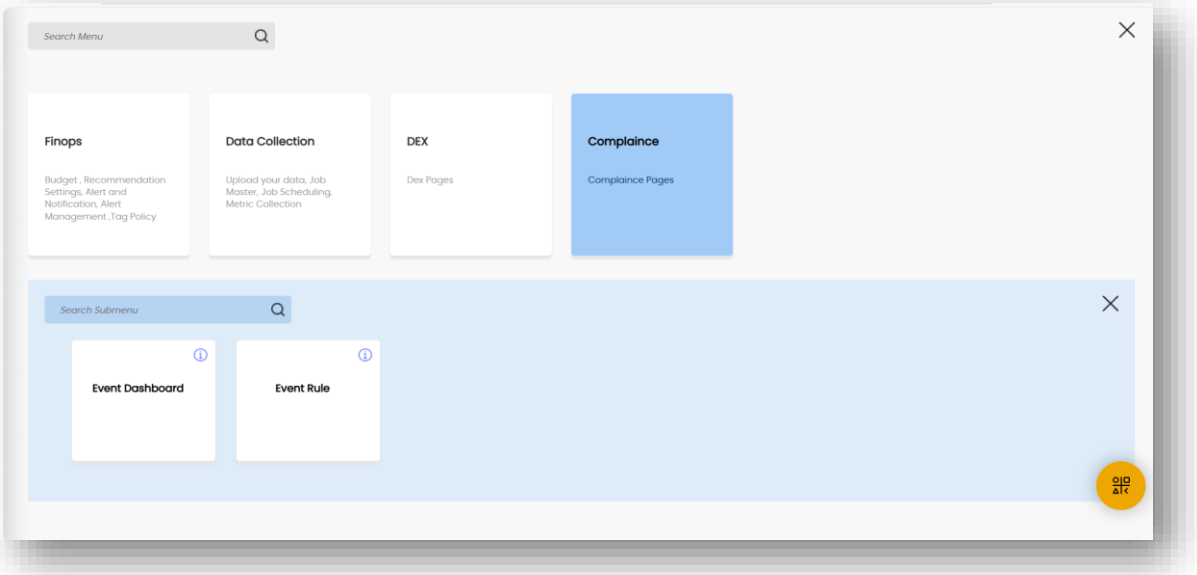

Figure 97 – Event Dashboard

2. The Event Dashboard lists the alerts created for Eventrule configured for Compliance.
3. As soon as the user clicks on EventDashboard, the user can see the screen below:

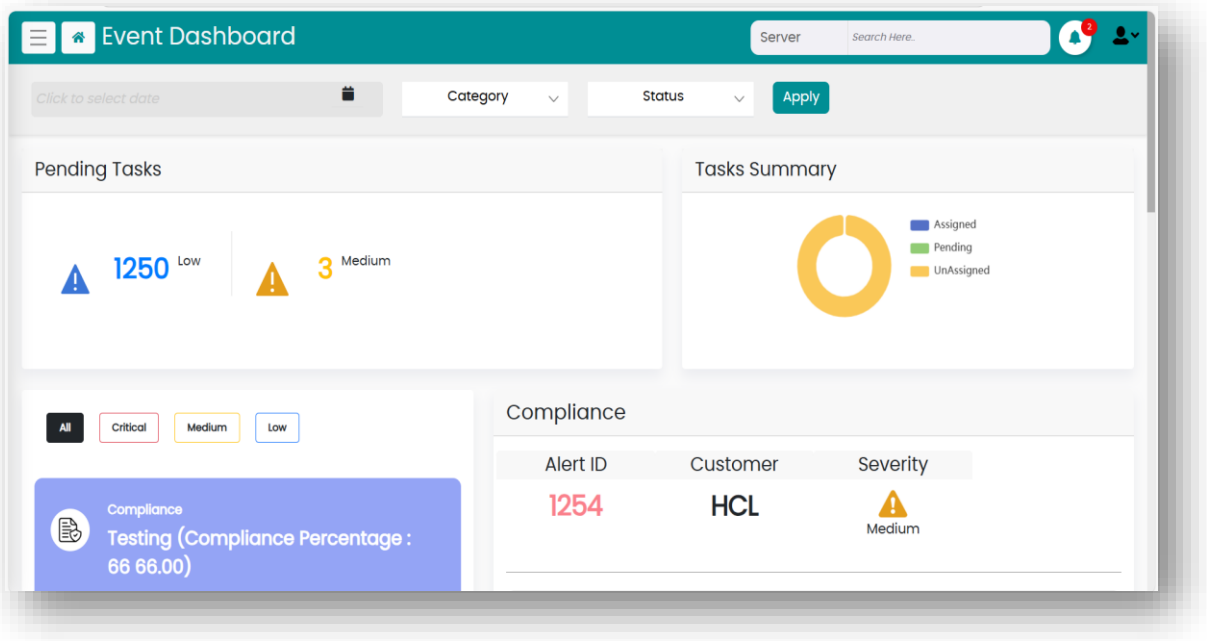

Figure 98 – Event Dashboard

4. In this dashboard, you can have a look at the alerts which are in:
   - Pending status severity wise.
   - Count of alerts status wise.
   - Users can filter alerts based on creation date and category:

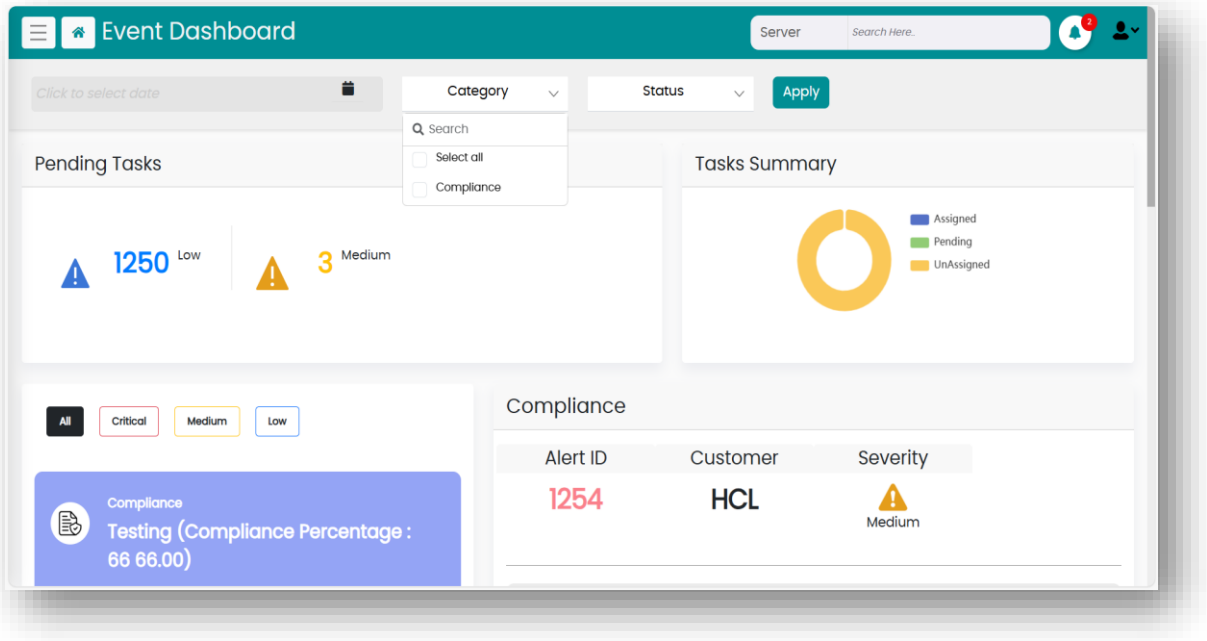

Figure 99 – Pending Task

5. By clicking on any alert on the left side, the details for that alert can be seen on the right-hand side as shown in the following figure.

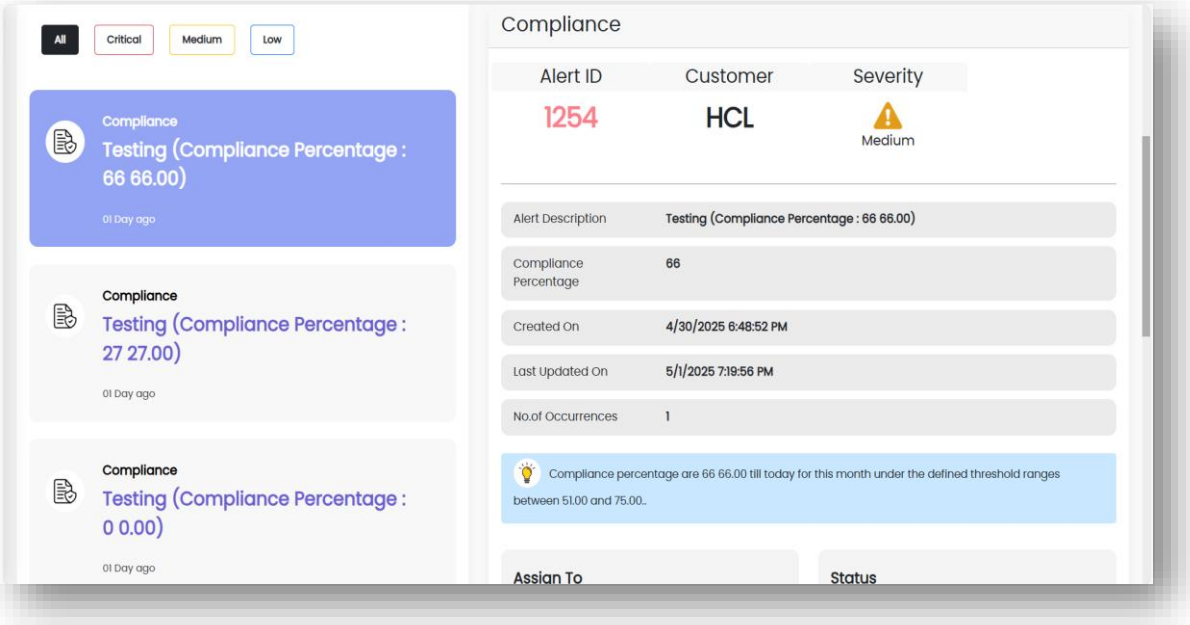

Figure 100 – Compliance Alert Details

6. In the detailed view the alert can be assigned to any assignee putting some remarks.

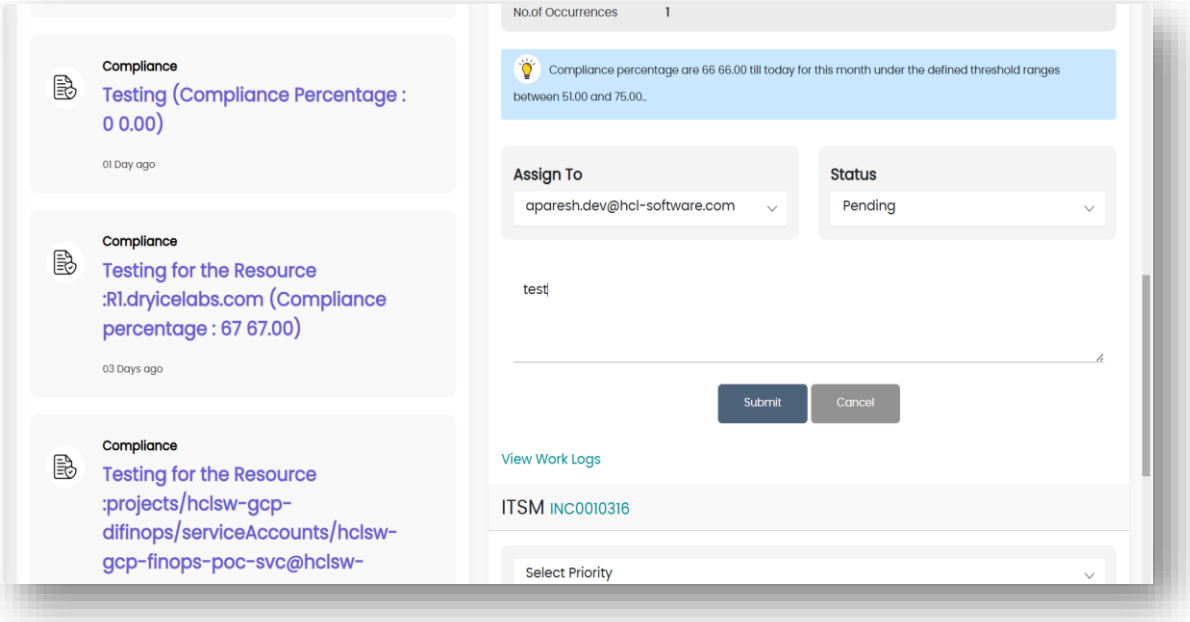

Figure 101 – Assigning

7. In the detailed view, there is a link View Work logs. This can be used to view the work logs done till date on the alert.

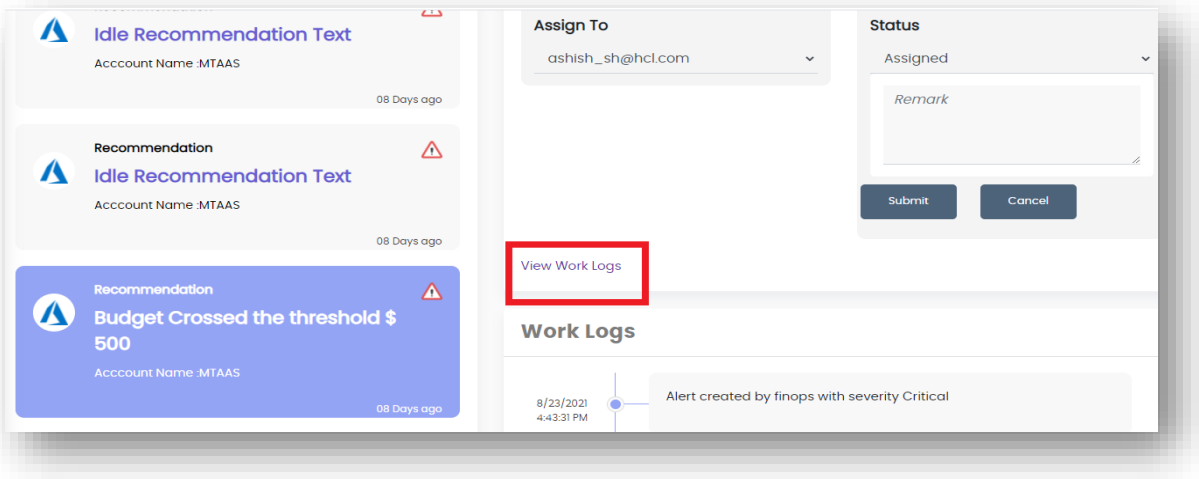

Figure 102 – Work Logs

8. In the ITSM section, User can create the servicenow ticket based on priority. Select the Priority and click on Create button.

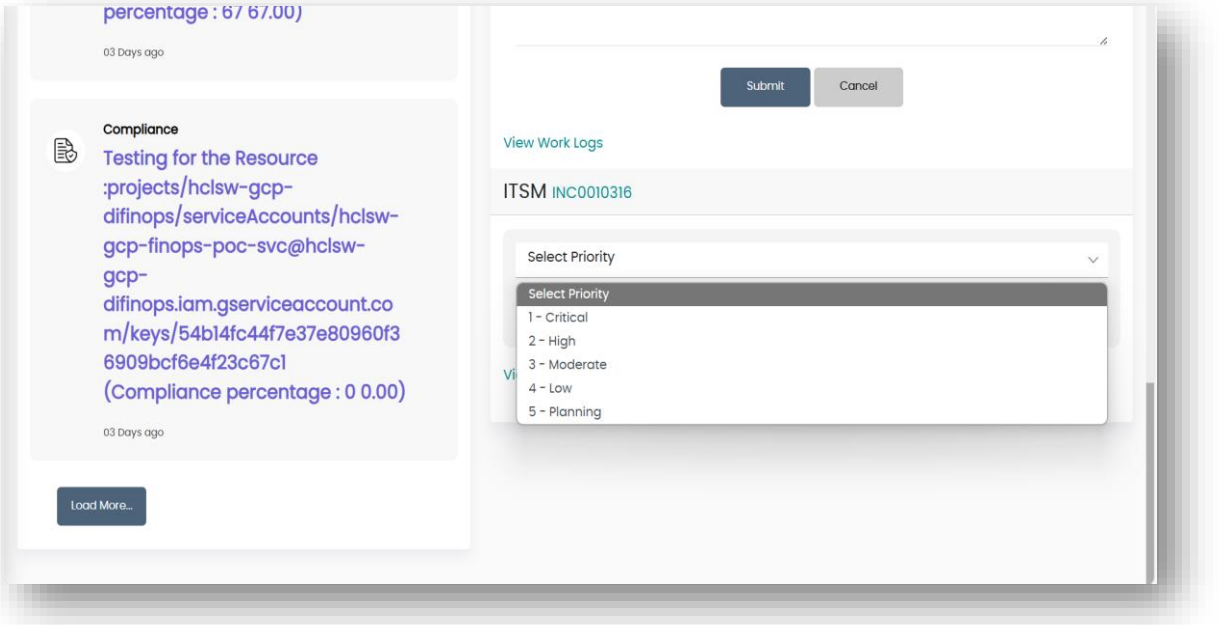

Figure 103 – Incident of ITSM

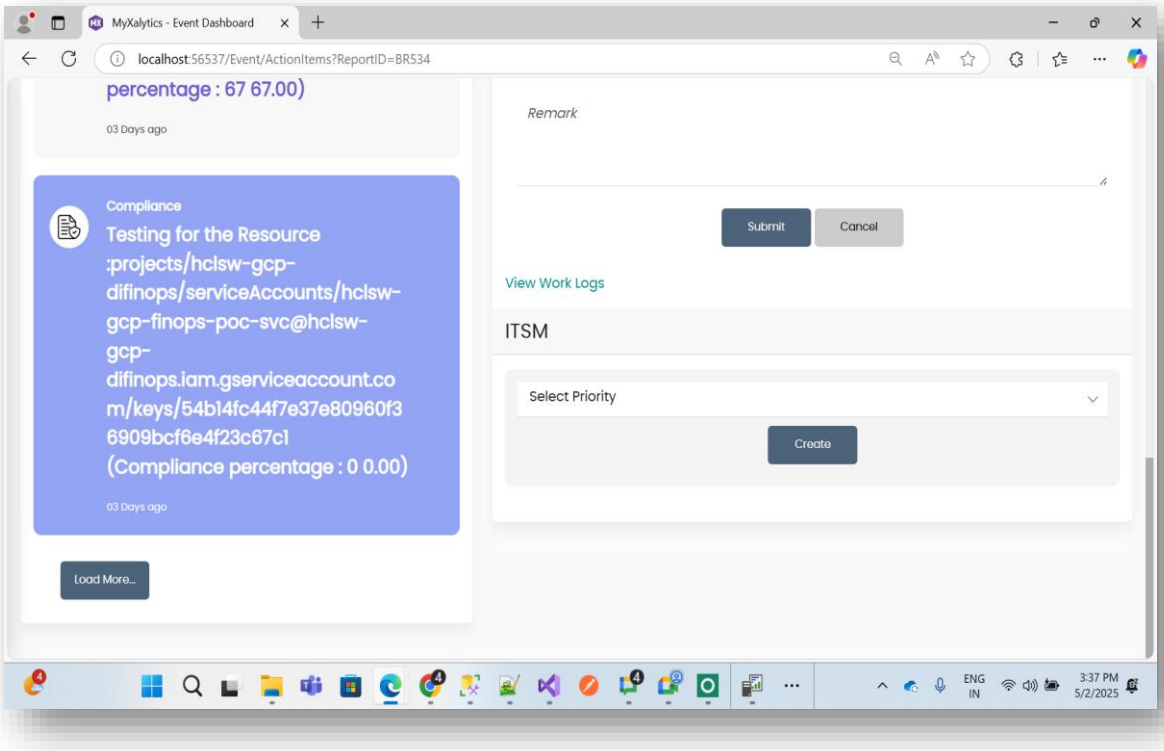

Figure 104 – Creating ITSM Ticket

6 Support

For any additional queries, please reach out to us at teammyxalyticsps@hcl-software.com.

HCLSoftware

hcltechsw.com