LATHA MATHAVAN ENGINEERING COLLEGE

KIDARIPATTI, ALAGARKOIL, MADURAI-625301.

CCS336-CLOUD SERVICE MANAGEMENT

Regulation : 2021

Branch : *B.E.* – CSE

Year :III - YEAR

Semester :VI- Semester

LATHA MATHAVAN ENGINEERING COLLEGE

KIDARIPATTI, ALAGARKOIL, MELUR TALUKMADURAI - 625301

Department of _____ _____ Engineering Laboratory Record

NAME: _____ CLASS_____

REGISTER NO: _____

Certified that this is bona fide record of work done by the above student of the **CCS336-Cloud Service Mangement lab** during the year _____

Signature of Lab-in-Charge

Signature of Head of the Department

Submitted for the Practical Examination Held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

CCS336-CLOUD SERVICES MANAGEMENT

COURSE OBJECTIVES:

- Introduce Cloud Service Management terminology, definition & concepts
- Compare and contrast cloud service management with traditional IT service management
- Identify strategies to reduce risk and eliminate issues associated with adoption of cloud services
- Select appropriate structures for designing, deploying and running cloud-based services in a business environment
- Illustrate the benefits and drive the adoption of cloud-based services to solve real world problems

PRACTICAL EXERCISES: 30 PERIODS

1. Create a Cloud Organization in AWS/Google Cloud/or any equivalent Open Source cloud softwares like Open stack, Eucalyptus, Open Nebula with Role-based access control
2. Create a Cost-model for a web application using various services and do Cost-benefit analysis
3. Create alerts for usage of Cloud resources
4. Create Billing alerts for your Cloud Organization
5. Compare Cloud cost for a simple web application across AWS, Azure and GCP and suggest the best one

COURSE OUTCOMES:

CO1: Exhibit cloud-design skills to build and automate business solutions using cloud technologies.

CO2: Possess Strong theoretical foundation leading to excellence and excitement towards adoption of cloud-based services

CO3: Solve the real world problems using Cloud services and technologies

Ex No:1 Creation of Cloud Organization in AWS/Google Cloud

AIM

To Create a Cloud Organization in AWS/Google Cloud/or any equivalent Open Source cloud softwares like Openstack, Eucalyptus, OpenNebula with Role-based access control

PROCEDURE

Google Cloud Platform (**GCP**), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, Google Drive, and YouTube Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. Registration requires a credit card or bank account details.[4]

Google Cloud Platform provides infrastructure as a service, platform as a service, and serverless computing environments.

In April 2008, Google announced App Engine, a platform for developing and hosting web applications in Google-managed data centers, which was the first cloud computing service from the company. The service became generally available in November 2011. Since the announcement of App Engine, Google added multiple cloud services to the platform.

Google Cloud Platform is a partof **Google Cloud**, which includes the Google Cloud Platform public cloud infrastructure, as well as Google Workspace (G Suite), enterprise versions of Android and ChromeOS, and application programming interfaces (APIs) for machine learning and enterprise mapping services.

Creating your project

bookmark_border

To deploy your app on App Engine, you must create a Google Cloud project, which is a top level container that holds your App Engine application resources as well as other Google Cloud resources.

In this task, you create a Google Cloud project and an App Engine application to store settings, computing resources, credentials, and metadata for your app.

If you already have a Google Cloud project with App Engine and the Cloud Build API enabled, continue to Writing Your Web Service.

Creating a Google Cloud project

1. If you're new to Google Cloud, create an account to evaluate how our products perform in real-world scenarios. New customers get $300 in free credits to run, test, and deploy workloads.
2. In the Google Cloud console, on the project selector page, select or create a Google Cloud project.

   Note: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

   Go to project selector
3. Make sure that billing is enabled for your Google Cloud project.

4. Enable the Cloud Build API.

   Enable the API

5. Install the Google Cloud CLI.

6. To initialize the gcloud CLI, run the following command:

   gcloud init

7. Create an App Engine application for your Google Cloud project in the Google Cloud console.

   Open app creation

8. Select a region where you want your app's computing resources located.

   Note: After you create your App Engine app, you cannot change the region. To reduce latency, choose the region closest to your app's intended users. For more information on the available regions, see App Engine Locations.

   Next step

   Now that your Google Cloud project is set up, you're ready to write a basic web service with Node.js.

Setting up your development environment

book mark_border

Go Java Node.js PHP Python Ruby

Use the following steps to set up your local environment for developing and deploying your App Engine services:

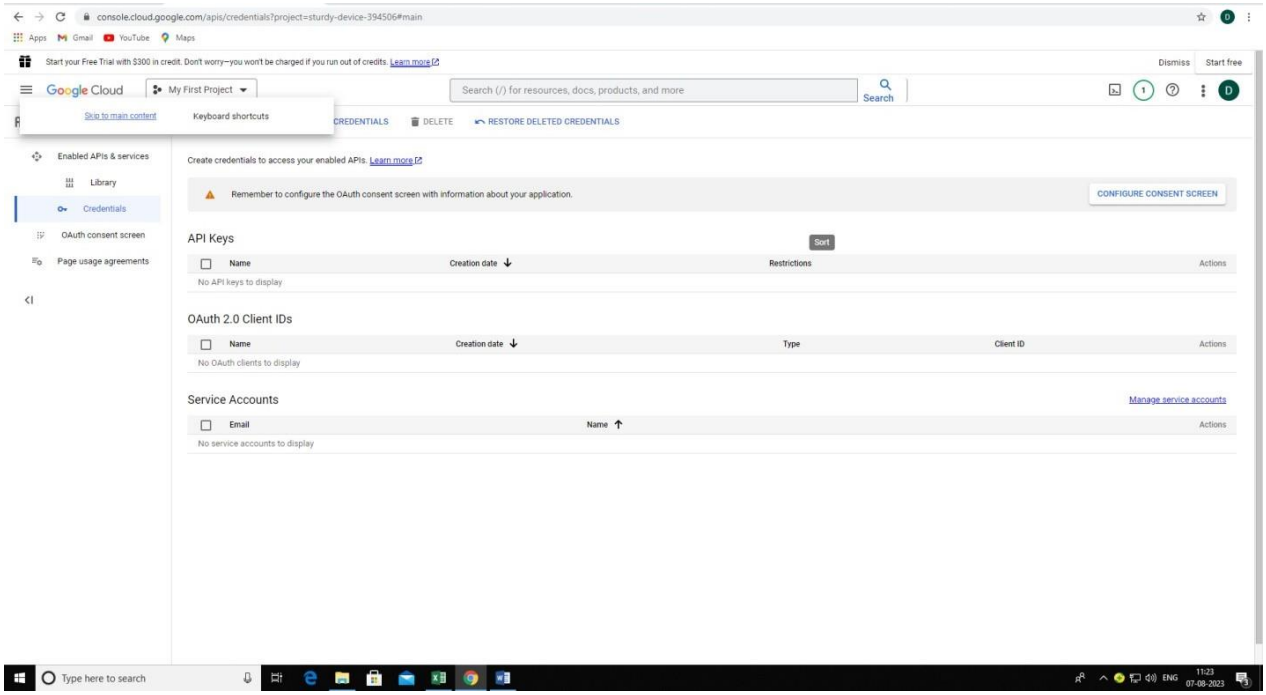
1. Install the latest release of Python 3.

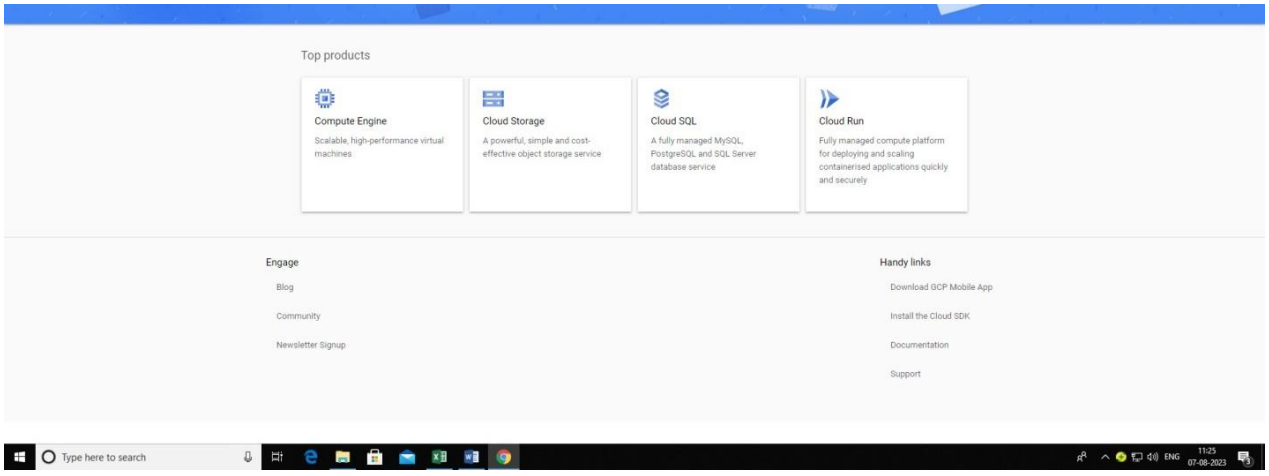
   See Python3 Runtime Environment for a list of the supported versions.

2. Install and initialize the gcloud CLI for deploying and managing your apps. If you already have the gcloud CLI installed and initialized, run the gcloud components update command to update to the latest release. By downloading, you agree to be bound by the Terms that govern use of the gcloud CLI for App Engine.

   Optional tools:

- Install Git for access to code, samples, libraries, and tools in the Google Cloud GitHub repository.
- Install your preferred tooling or framework, for example you can use any of the following frameworks to develop your Python 3 app:
- Flask
- Django
- Pyramid
- Bottle
- web.py
- Tornado





Result:

Thus Cloud Organization in AWS/Google Cloud/or any equivalent Open Source cloud softwares like Openstack, Eucalyptus, Open Nebula with Role-based access control was created.

Ex No:2 Creation of Cost-model for a web application

AIM:

To Create a Cost-model for a web application using various services and do Cost-benefit analysis.

PROCEDURE:

Cost modeling is an exercise where you create logical groups of cloud resources that are mapped to the organization's hierarchy and then estimate costs for those groups. The goal of cost modeling is to estimate the overall cost of the organization in the cloud.

Understand how your responsibilities align with your organization

Map your organization's needs to logical groupings offered by cloud services. This way the business leaders of the company get a clear view of the cloud services and how they're controlled.

Capture clear requirements

Start your planning by carefully enumerating the requirements. From the high-level requirements, narrow down each requirement before starting on the design of the solution.

Consider the cost constraints

Evaluate the budget constraints on each business unit. Determine the governance policies in Azure to lower cost by reducing wastage, over-provisioning, or expensive provisioning of resources.

Consider tradeoffs

Optimal design doesn't equate to a lowest-cost design.

As you prioritize requirements, cost can be adjusted. Expect a series of tradeoffs in the areas that you want to optimize, such as security, scalability, resilience, and operability. If the cost to address the challenges in those areas is high, stakeholders look for alternate options to reduce cost. There might be risky choices made in favor of a cheaper solution.

Derive functional requirements from high-level goals

Break down the high-level goals into functional requirements for the solution's components. Each requirement must be based on realistic metrics to estimate the actual cost of the workload.

Consider the billing model for Azure resources

Azure services are offered with consumption-based prices where you're charged for only what you use. There's also options for fixed price where you're charged for provisioned resources.

Most services are priced based on units of size, amount of data, or operations. Understand the meters that are available to track usage. For more information, see Azure resources.

At the end of this exercise, you should have identified the lower and upper limits on cost and set budgets for the workload. Azure lets you create and manage budgets in Azure Cost Management.

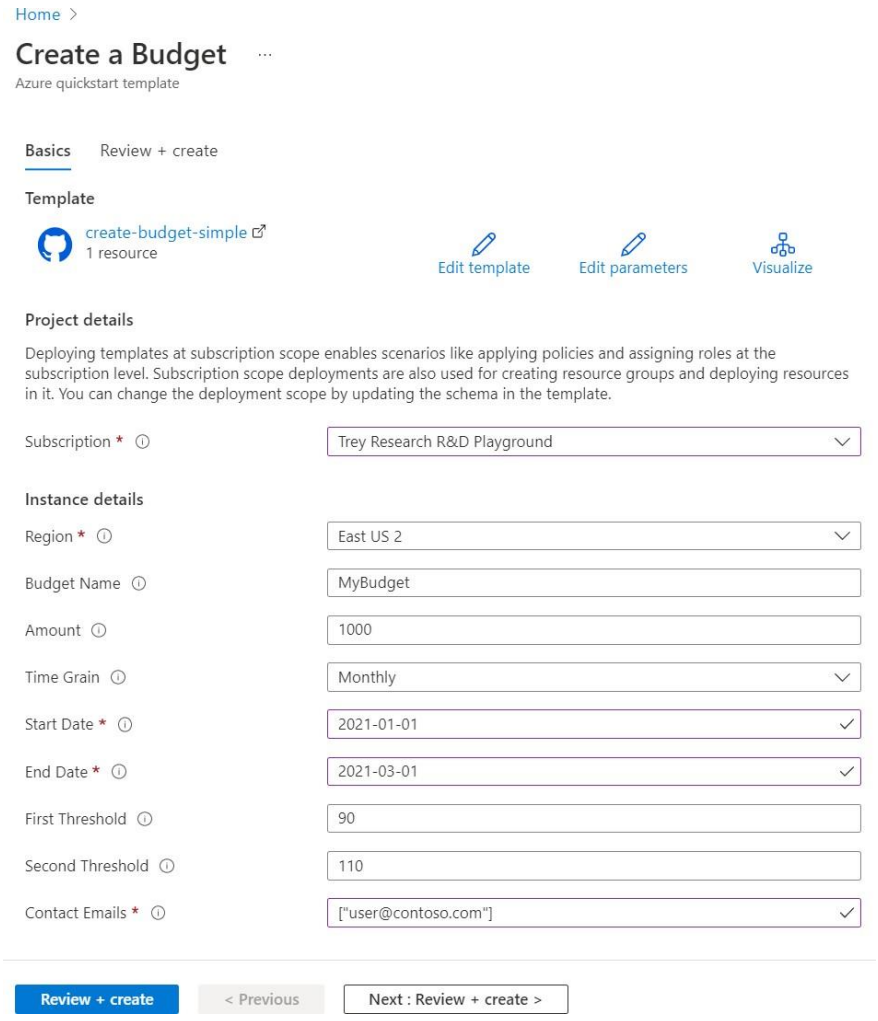
Budgets are supported for the following types of Azure account types and scopes:

- Azure role-based access control (Azure RBAC) scopes
  - Management groups
  - Subscription
- Enterprise Agreement scopes
  - Billing account
  - Department
  - Enrollment account
- Individual agreements
  - Billing account
- Microsoft Customer Agreement scopes
  - Billing account
  - Billing profile
  - Invoice section
  - Customer
- AWS scopes
  - External account
  - External subscription

Deploy the template

1. Select the following image to sign in to Azure and open a template. The template creates a budget without any filters

2. Select or enter the following values.



- **Subscription**: select an Azure subscription.
- **Resource group**: if required, select an existing resource group, or **Create new**.
- **Region**: select an Azure region. For example, **Central US**.
- **Budget Name**: enter a name for the budget. It should be unique within a resource group. Only alphanumeric, underscore, and hyphen characters are allowed.
- **Amount**: enter the total amount of cost to track with the budget.
- **Time Grain**: enter the time covered by a budget. Allowed values are Monthly, Quarterly, or Annually. The budget resets at the end of the time grain.
- **Start Date**: enter the start date with the first day of the month in YYYY-MM-DD format. A future start date shouldn't be more than three months from today.

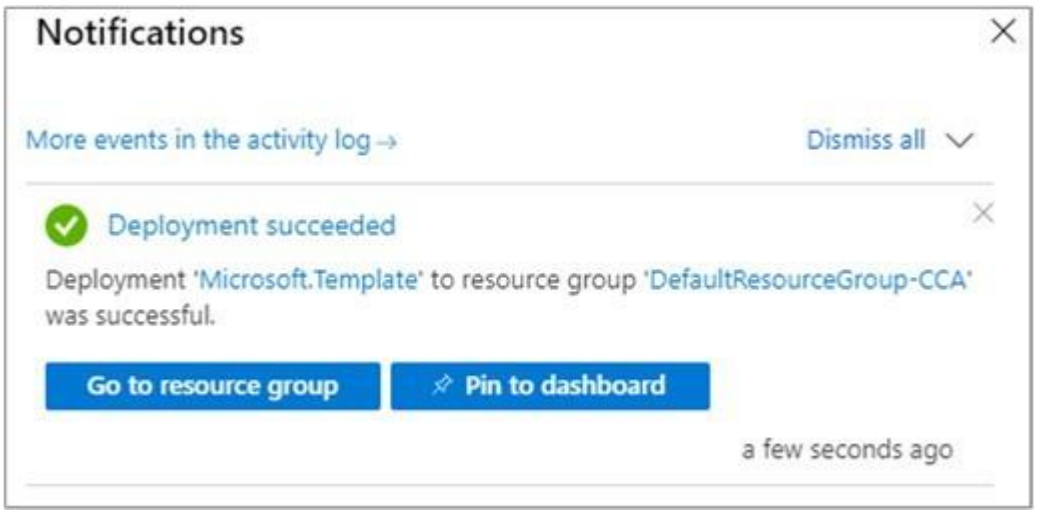
- You can specify a past start date with the Time Grain period.
- **End Date**: enter the end date for the budget in YYYY-MM-DD format.
- **First Threshold**: enter a threshold value for the first notification. A notification is sent when the cost exceeds the threshold. It's always percent and has to be between 0.01 and 1000.
- **Second Threshold**: enter a threshold value for the second notification. A notification is sent when the cost exceeds the threshold. It's always percent and has to be between 0.01 and 1000.
- **Contact Emails** enter a list of email addresses to send the budget notification to when a threshold is exceeded. It accepts an array of strings. Expected format is ["user1@domain.com","user2@domain.com"].

Depending on your Azure subscription type, do one of the following actions:

- Select **Review + create**.

- Review the terms and conditions, select **I agree to the terms and conditions stated above**, and then select **Purchase**.

If you selected **Review + create**, your template is validated. Select **Create**.



The Azure portal is used to deploy the template. In addition to the Azure portal, you can also use Azure PowerShell, Azure CLI, and REST API. To learn about other deployment templates, see Deploy templates.

Validate the deployment

Use one of the following ways to verify that the budget is created.

- Azure portal
- CLI
- PowerShell

Navigate to **Cost Management + Billing** > select a scope > **Budgets**.

Clean up resources

When you no longer need a budget, delete it by using one the following methods:

- Azure portal
- CLI
- PowerShell

Navigate to **Cost Management + Billing** > select a billing scope > **Budgets** > select a budget > then select **Delete budget**.

RESULT:

Thus a Cost-model for a web application using various services and do Cost-benefit analysis was created

Ex No:3 **Create alerts for usage of Cloud resources**

AIM:

To create alerts for usage of Cloud resources.

PROCEDURE:

Azure Monitor alerts

Azure Monitor offers alerting capabilities to notify you, via email or messaging, when things go wrong. These capabilities are based on a common data-monitoring platform that includes logs and metrics generated by your servers and other resources. By using a common set of tools in Azure Monitor, you can analyze data that's combined from multiple resources and use it to trigger alerts. These triggers can include:

- Metric values.
- Log search queries.
- Activity log events.
- The health of the underlying Azure platform.
- Tests for website availability.

See the list of Azure Monitor data sources for a more detailed description of the sources of monitoring data that this service collects.

For details about manually creating and managing alerts by using the Azure portal, see the Azure Monitor documentation.

Automated deployment of recommended alerts

In this guide, we recommend that you create a set of 15 alerts for basic infrastructure monitoring. Find the deployment scripts in the Alert Toolkit GitHub repository.

This package creates alerts for:

- Low disk space
- Low available memory
- High CPU use
- Unexpected shutdowns
- Corrupted file systems
- Common hardware failures

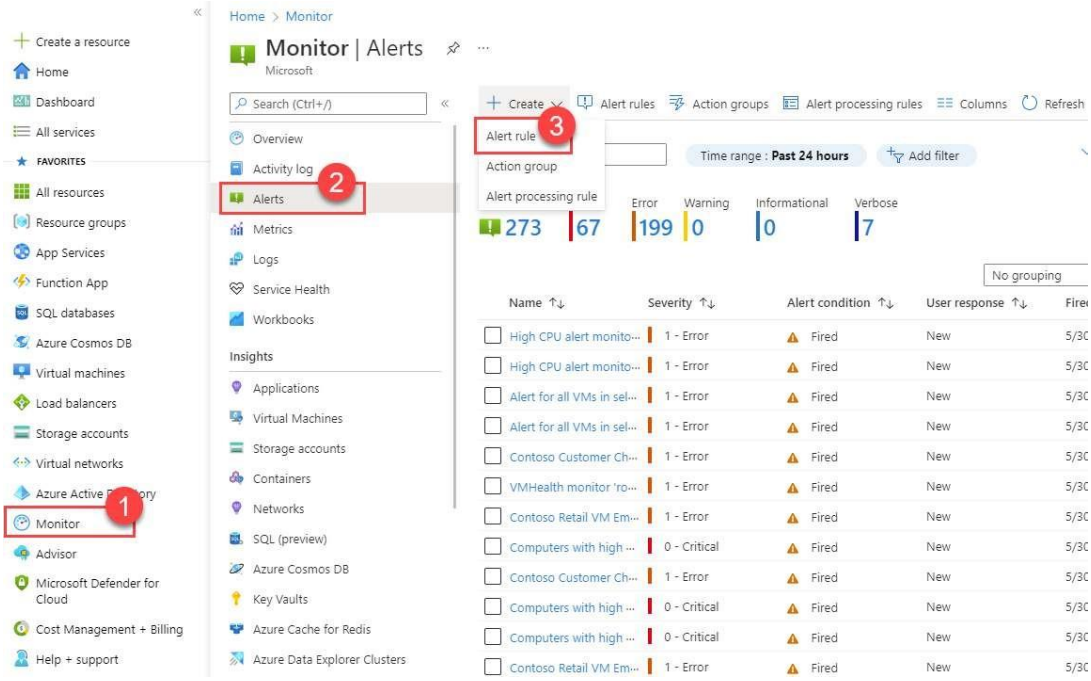
The package uses HPE server hardware as an example. Change the settings in the associated configuration file to reflect your OEM hardware. You can also add more performance counters to the configuration file. To deploy the package, run the New-CoreAlerts.ps1 file.

Create or edit an alert rule in the Azure portal

There are several ways that you can create a new alert rule. To create a new alert rule from the portal home page:

1. In the portal, select **Monitor** > **Alerts**.

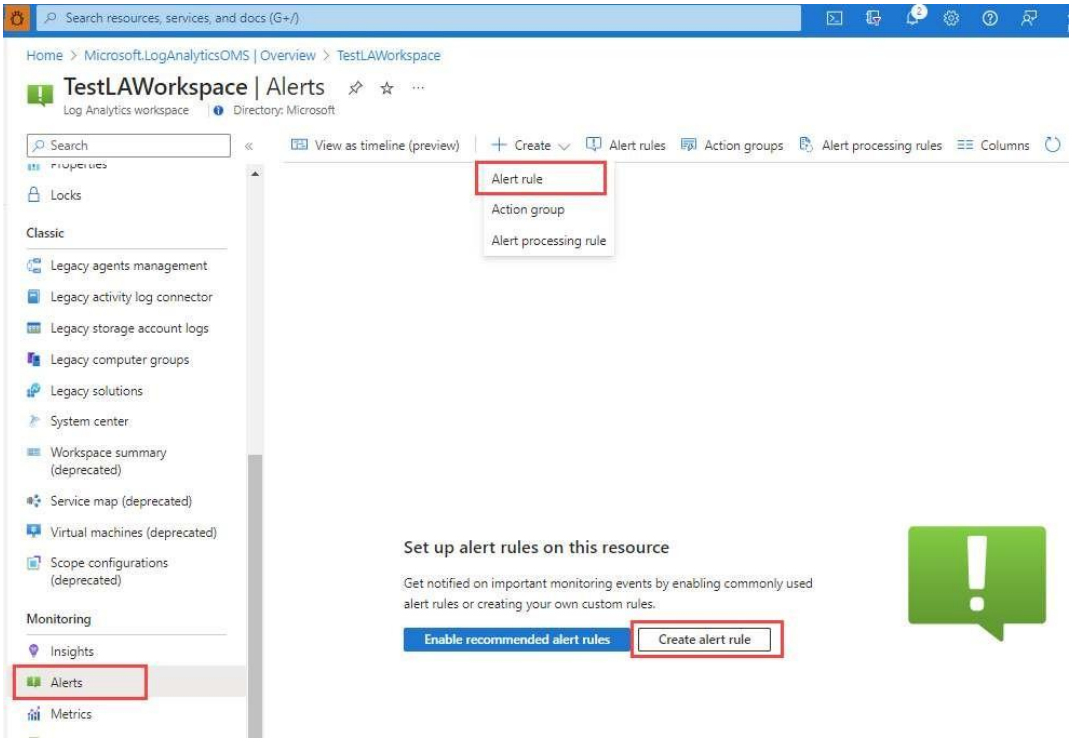
2. Open the **+ Create** menu, and select **Alert rule**.



To create a new alert rule from a specific resource:

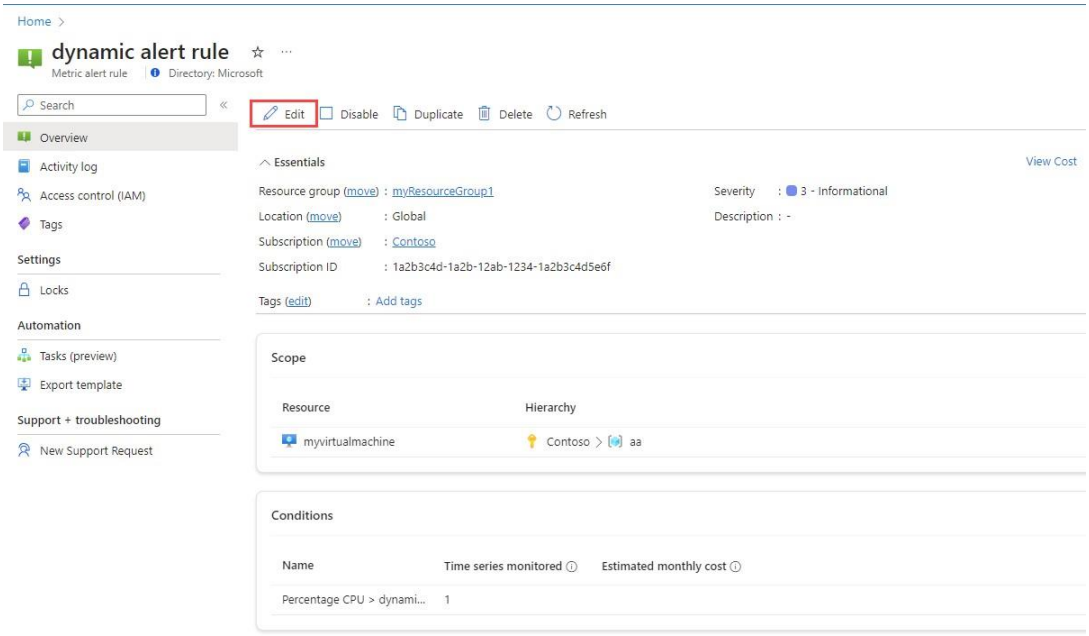
1. In the portal, navigate to the resource.

2. Select **Alerts** from the left pane, and then select **+ Create** > **Alert rule**.



To edit an existing alert rule:

1. In the portal, either from the home page or from a specific resource, select **Alerts** from the left pane.
2. Select **Alert rules**.
3. Select the alert rule you want to edit, and then select **Edit**.



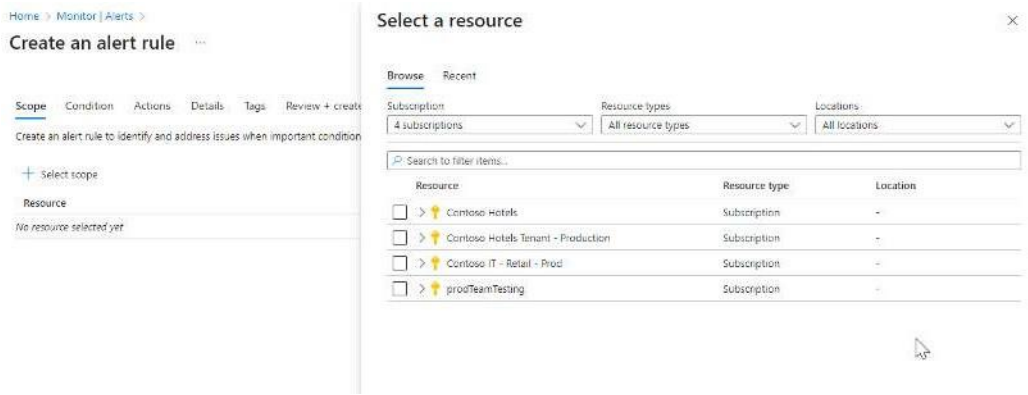
4. Select any of the tabs for the alert rule to edit the settings.

Set the alert rule scope

1. On the **Select a resource** pane, set the scope for your alert rule. You can filter by **subscription**, **resource type**, or **resource location**.

2. Select **Apply**.

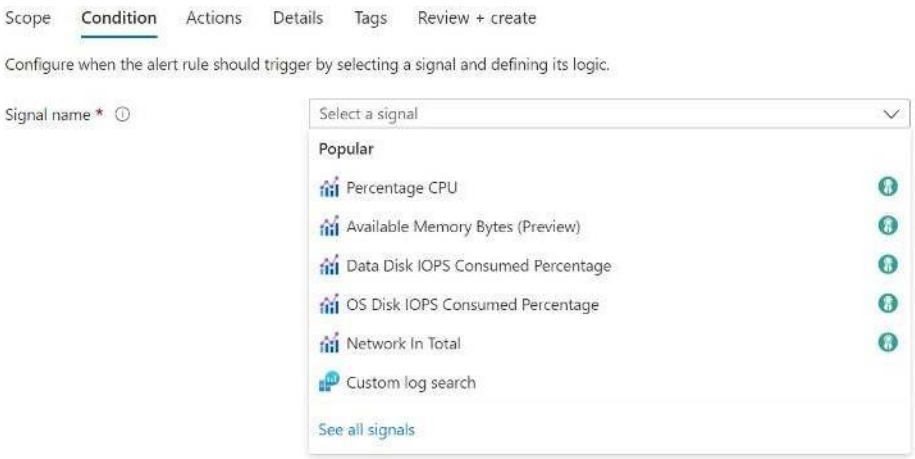
3. Select **Next: Condition** at the bottom of the page.



Set the alert rule conditions

1. On the **Condition** tab, when you select the **Signal name** field, the most commonly used signals are displayed in the drop-down list. Select one of these popular signals,

or select **See all signals** if you want to choose a different signal for the condition.



2. (Optional) If you chose to **See all signals** in the previous step, use the **Select a signal** pane to search for the signal name or filter the list of signals. Filter by:

   - **Signal type**: The type of alert rule you're creating.
   - **Signal source**: The service sending the signal. The list is prepopulated based on the type of alert rule you selected.

   This table describes the services available for each type of alert rule:

Signal type	Signal source	Description
Metrics	Platform	For metric signals, the monitor service is the metric namespace. "Platform" means the metrics are provided by the resource provider, namely, Azure.
	Azure.ApplicationInsights	Customer-reported metrics, sent by the Application Insights SDK.
	Azure.VM.Windows.GuestMetrics	VM guest metrics, collected by an extension running on the VM. Can include built-in operating system perf counters and custom perf counters.
	<your custom namespace>	A custom metric namespace, containing custom metrics sent with the Azure Monitor Metrics API.
Log	Log Analytics	The service that provides the "Custom log search" and "Log (saved query)" signals.
Activity log	Activity log – Administrative	The service that provides the Administrative activity log events.
	Activity log – Policy	The service that provides the Policy activity log events.
	Activity log – Autoscale	The service that provides the Autoscale activity log events.
	Activity log – Security	The service that provides the Security activity log events.
Resource health	Resource health	The service that provides the resource-level health status.
Service health	Service health	The service that provides the subscription-level health status.

Select the **Signal name** and **Apply**.

Follow the steps in the tab that corresponds to the type of alert you're creating.

a. Preview the results of the selected metric signal in the **Preview** section. Select values for the following fields.

Field	Description
Time range	The time range to include in the results. Can be from the last six hours to the last week.
Time series	The time series to include in the results.

b. In the **Alert logic** section:

Field	Description
Threshold	Select if the threshold should be evaluated based on a static value or a dynamic value. A **static threshold** evaluates the rule by using the threshold value that you configure. **Dynamic thresholds** use machine learning algorithms to continuously learn the metric behavior patterns and calculate the appropriate thresholds for unexpected behavior. You can learn more about using dynamic thresholds for metric alerts.
Operator	Select the operator for comparing the metric value against the threshold. If you're using dynamic thresholds, alert rules can use tailored thresholds based on metric behavior for both upper and lower bounds in the same alert rule. Select one of

Field	Description
	these operators: - Greater than the upper threshold or lower than the lower threshold (default) - Greater than the upper threshold - Lower than the lower threshold
Aggregation type	Select the aggregation function to apply on the data points: Sum, Count, Average, Min, or Max.
Threshold value	If you selected a **static** threshold, enter the threshold value for the condition logic.
Unit	If the selected metric signal supports different units, such as bytes, KB, MB, and GB, and if you selected a **static** threshold, enter the unit for the condition logic.
Threshold sensitivity	If you selected a **dynamic** threshold, enter the sensitivity level. The sensitivity level affects the amount of deviation from the metric series pattern that's required to trigger an alert. - **High**: Thresholds are tight and close to the metric series pattern. An alert rule is triggered on the smallest deviation, resulting in more alerts. - **Medium**: Thresholds are less tight and more balanced. There are fewer alerts than with high sensitivity (default). - **Low**: Thresholds are loose, allowing greater deviation from the metric series pattern. Alert rules are only triggered on large deviations, resulting in fewer alerts.
Aggregation granularity	Select the interval that's used to group the data points by using the aggregation type function. Choose an **Aggregation granularity** (period) that's greater than the **Frequency of evaluation** to reduce the likelihood of missing the first evaluation period of an added time series.
Frequency of evaluation	Select how often the alert rule is to be run. Select a frequency that's smaller than the aggregation granularity to generate a sliding window for the evaluation.

c. (Optional) Depending on the signal type, you might see the **Split by dimensions** section.

Dimensions are name-value pairs that contain more data about the metric value. By using dimensions, you can filter the metrics and monitor specific time-series, instead of monitoring the aggregate of all the dimensional values.

If you select more than one dimension value, each time series that results from the combination triggers its own alert and is charged separately. For example, the transactions metric of a storage account can have an API name dimension that contains the name of the API called by each transaction (for example, GetBlob, DeleteBlob, and PutPage). You can choose to have an alert fired when there's a high number of transactions in a specific API (the aggregated data). Or you can use dimensions to alert only when the number of transactions is high for specific APIs.

Field	Description
Dimension name	Dimensions can be either number or string columns. Dimensions are used to monitor specific time series and provide context to a fired alert. Splitting on the **Azure Resource ID** column makes the specified resource into the alert target. If detected, the **ResourceID** column is selected automatically and changes the context of the fired alert to the record's resource.
Operator	The operator used on the dimension name and value.
Dimension values	The dimension values are based on data from the last 48 hours. Select **Add custom value** to add custom dimension values.
Include all future values	Select this field to include any future values added to the selected dimension.

d. (Optional) In the **When to evaluate** section:

Field	Description
Check every	Select how often the alert rule checks if the condition is met.
Lookback period	Select how far back to look each time the data is checked. For example, every 1 minute, look back 5 minutes.

e. (Optional) In the **Advanced options** section, you can specify how many failures within a specific time period trigger an alert. For example, you can specify that you only want to trigger an alert if there were three failures in the last hour. Your application business policy should determine this setting.

Select values for these fields:

Field	Description
Number of violations	The number of violations within the configured time frame that trigger the alert.
Evaluation period	The time period within which the number of violations occur.
Ignore data before	Use this setting to select the date from which to start using the metric historical data for calculating the dynamic thresholds. For example, if a resource was running in testing mode and is moved to production, you may want to disregard the metric behavior while the resource was in testing.

f. Select **Done**.

From this point on, you can select the **Review + create** button at any time.

Set the alert rule actions

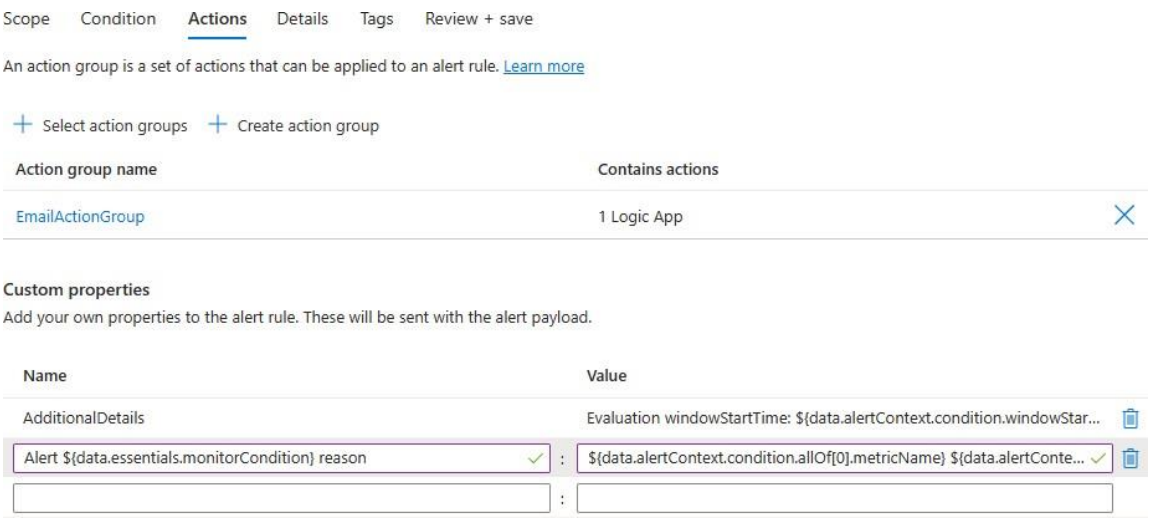
1. On the **Actions** tab, select or create the required action groups.

2. (Optional) In the **Custom properties** section, if you've configured action groups for this alert rule, you can add your own properties to include in the alert notification payload. You can use these properties in the actions called by the action group, such as webhook, Azure function or logic app actions.

   The custom properties are specified as key:value pairs, using either static text, a dynamic value extracted from the alert payload, or a combination of both.

   The format for extracting a dynamic value from the alert payload is: ${<path to schema field>}. For example: ${data.essentials.monitorCondition}.

   Use the common alert schema format to specify the field in the payload, whether or not the action groups configured for the alert rule use the common schema.



In the following examples, values in the **custom properties** are used to utilize data from a payload that uses the common alert schema:

Example 1

This example creates an "Additional Details" tag with data regarding the "window start time" and "window end time".

- **Name:** "Additional Details"
- **Value:** "Evaluation windowStartTime: ${data.context.condition.windowStartTime}. windowEndTime: ${data.context.condition.windowEndTime}"
- **Result:** "AdditionalDetails:Evaluation windowStartTime: 2023-04-04T14:39:24.492Z. windowEndTime: 2023-04-04T14:44:24.492Z"

Example 2 This example adds the data regarding the reason of resolving or firing the alert.

- **Name:** "Alert ${data.essentials.monitorCondition} reason"
- **Value:** "${data.context.condition.allOf[0].metricName} ${data.context.condition.allOf[0].operator} ${data.context.condition.allOf[0].threshold} ${data.essentials.monitorCondition}. The value is ${data.context.condition.allOf[0].metricValue}"
- **Result:** Example results could be something like:

    - "Alert Resolved reason: Percentage CPU GreaterThan5 Resolved. The value is 3.585"
    - “Alert Fired reason": "Percentage CPU GreaterThan5 Fired. The value is 10.585"

Note

The **common schema** overwrites custom configurations. Therefore, you can't use both custom properties and the common schema for log alerts.

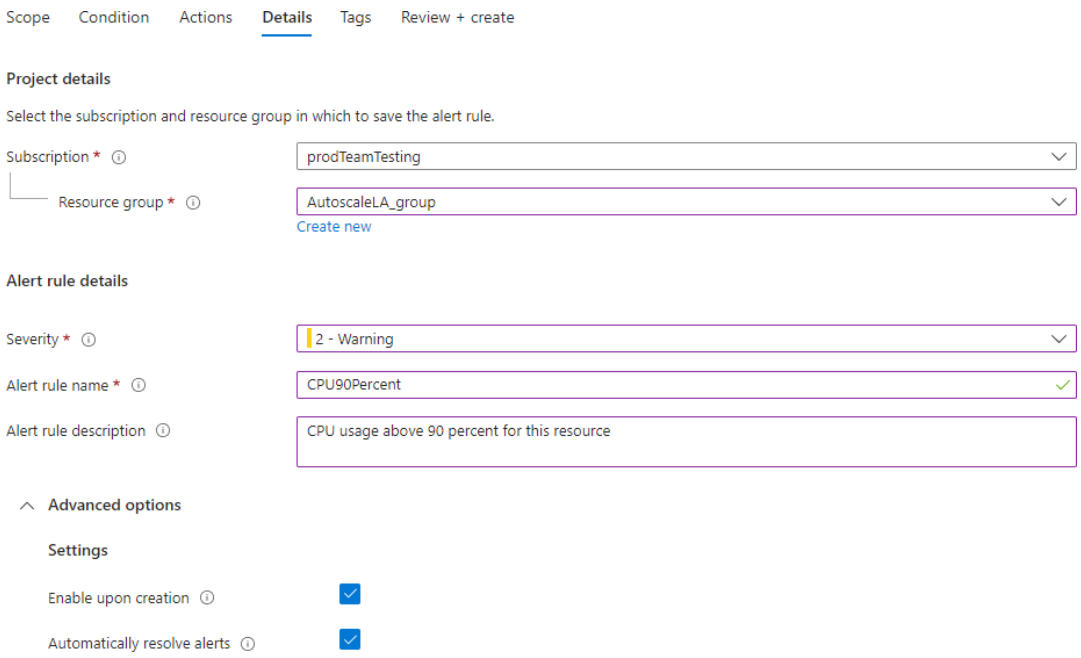
Set the alert rule details

1. On the **Details** tab, define the **Project details**.

    - Select the **Subscription**.
    - Select the **Resource group**.

Define the **Alert rule details**.

- Metric alert
- Log alert
- Activity log alert
- Resource Health alert
- Service Health alert



f. Select the **Severity**.

g. Enter values for the **Alert rule name** and the **Alert rule description**.

h. (Optional) If you're creating a metric alert rule that monitors a custom metric with the scope defined as one of the following regions and you want to make sure that the data processing for the alert rule takes place within that region, you can select to process the alert rule in one of these regions:

    - North Europe
    - West Europe
    - Sweden Central
    - Germany West Central

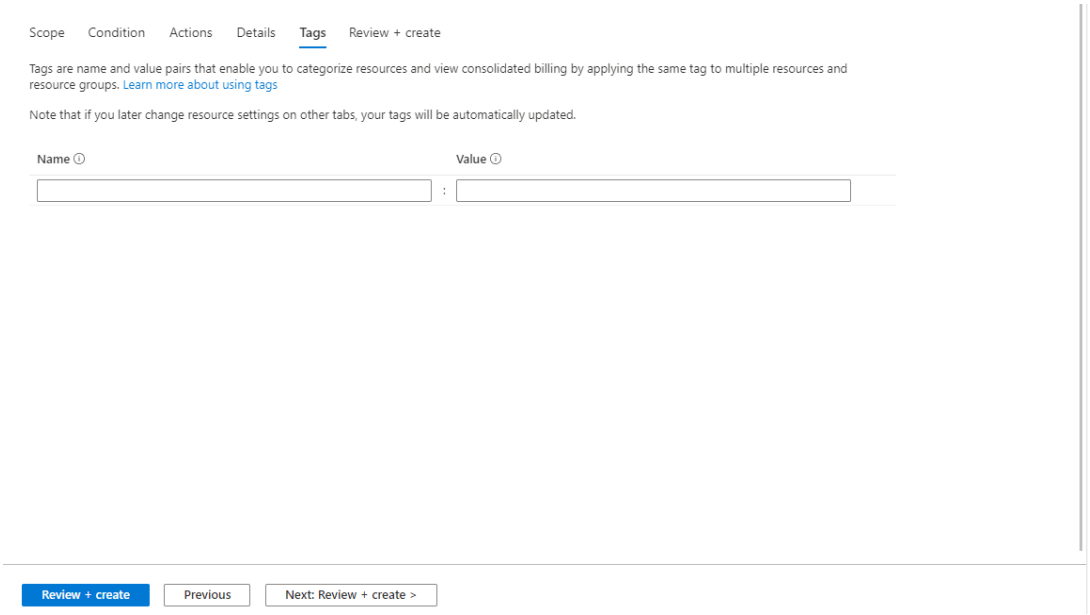
    We're continually adding more regions for regional data processing.

i. (Optional) In the **Advanced options** section, you can set several options.

Field	Description
Enable upon creation	Select for the alert rule to start running as soon as you're done creating it.
Automatically resolve alerts (preview)	Select to make the alert stateful. When an alert is stateful, the alert is resolved when the condition is no longer met. If you don't select this checkbox, metric alerts are stateless. Stateless alerts fire each time the condition is met, even if alert already fired. The frequency of notifications for stateless metric alerts differs based on the alert rule's configured frequency: **Alert frequency of less than 5 minutes**: While the condition continues to be met, a notification is sent somewhere between one and six minutes. **Alert frequency of more than 5 minutes**: While the condition continues to be met, a notification is sent between the configured frequency and double the value of the frequency. For example, for an alert rule with a frequency of 15 minutes, a notification is sent somewhere between 15 to 30 minutes.

Finish creating the alert rule

1. On the **Tags** tab, set any required tags on the alert rule resource.



2. On the **Review + create** tab, the rule is validated, and lets you know about any issues.

3. When validation passes and you've reviewed the settings, select the **Create** button.

Product Details

Log alerts

1 Condition

Terms of use | Privacy statement

Total Total pricing

$

Pricing

Scope

Resource ACME-Portal > ACME Telco > MyResourceGro...

Condition

Search query

Measure	Table rows
Aggregation type	Count
Aggregation granularity	5 minutes

Alert logic

Operator	GreaterThan
Threshold value	5
Frequency of evaluation	5 minutes

Advanced options

Create Previous

RESULT:

Thus the alerts for usage of Cloud resources were created.

EX.NO.4 Creation of Billing alerts for your Cloud Organization

AIM:

To Create Billing alerts for your Cloud Organization.

PROCEDURE:

Requirements

- You must be signed in using account root user credentials or as an IAM user that has been given permission to view billing information.
- For consolidated billing accounts, billing data for each linked account can be found by logging in as the paying account. You can view billing data for total estimated charges and estimated charges by service for each linked account, in addition to the consolidated account.
- In a consolidated billing account, member linked account metrics are captured only if the payer account enables the Receive Billing Alerts preference. If you change which account is your management/payer account, you must enable the billing alerts in the new management/payer account.
- The account must not be part of the Amazon Partner Network (APN) because billing metrics are not published to Cloud Watch for APN accounts. For more information, see AWS Partner Network.To enable the monitoring of estimated charges

1. Open the AWS Billing console at https://console.aws.amazon.com/billing/.
2. In the navigation pane, choose Billing Preferences.
3. By Alert preferences choose Edit.
4. Choose Receive Cloud Watch Billing Alerts.
5. Choose Save preferences.

Create a billing Alarm

Before you create a billing alarm, you must set your Region to US East (N. Virginia). Billing metric data is stored in this Region and represents worldwide charges. You also must enable billing alerts for your account or in the management/payer account (if you are using consolidated billing). For more information, see Enabling billing alerts.

In this procedure, you create an alarm that sends a notification when your estimated charges for AWS exceed a defined threshold.

To create a billing alarm using the Cloud Watch console

1. Open the Cloud Watch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose Alarms, and then choose All alarms.

3. Choose Create alarm.
4. Choose Select metric. In Browse, choose Billing, and then choose Total Estimated Charge.
   Note

   If you dont't see the Billing/Total Estimated Charge metric, enable billing alerts, and change your Region to US East. For more information, see Enabling billing alerts.
5. Select the box for the Estimated Charges metric, and then choose Select metric.
6. For Statistic, choose Maximum.
7. For Period, choose 6 hours.
8. For Threshold type, choose Static.
9. For Whenever Estimated Charges is . . ., choose Greater.
10. For than . . ., define the value that you want to cause your alarm to trigger. For example, 200 USD.

    The Estimated Charges metric values are only in US dollars (USD), and the currency conversion is provided by Amazon Services LLC. For more information, see What is AWS Billing?.
    Note

    After you define a threshold value, the preview graph displays your estimated charges for the current month.
11. Choose Additional Configuration and do the following:

- For Data points to alarm, specify 1 out of 1.
- For Missing data treatment, choose Treat missing data as missing.

12. Choose Next.
13. Under Notification, ensure that In alarm is selected. Then specify an Amazon SNS topic to be notified when your alarm is in the ALARM state. The Amazon SNS topic can include your email address so that you recieve email when the billing amount crosses the threshold that you specified.

    You can select an existing Amazon SNS topic, create a new Amazon SNS topic, or use a topic ARN to notify other account. If you want your alarm to send multiple notifications for the same alarm state or for different alarm states, choose Add notification.
14. Choose Next.

15. Under Name and description, enter a name for your alarm. The name must contain only UTF-8 characters, and can't contain ASCII control characters.

    a. (Optional) Enter a description of your alarm. The description can include markdown formatting, which is displayed only in the alarm Details tab in the Cloud Watch console. The markdown canbe useful to add links to runbooks or other internal resources.

    Choose Next.

    Under Preview and create, make sure that your configuration is correct, and then choose Create alarm.

Step 1: Go to the Cloud Watch console

https://console.aws.aazon.com/cloudwatch/

Step 2: Change the region

In the upper right hand corner, make sure your AWS region is set to US East (N. Virginia). This is the region where all billing data is stored.

Step 3: Create an alarm

Click on “Alarms” in the left side panel and in the dashboard that pops up click on the orange button that says “Create Alarm”.

Step 4: Select a metric

In the panel that pops up, click the button that says “select metric”. This will pop up a window that allows you to choose which type of service you’d like to use. Select “Billing”.

Step 5: Select the service

In the next window, you can choose to either base the alarm off a given service’s charges or the Total Estimated Charge on your account. We will set an alarm for our EC2 charges, so click “By Service”.

On the next step, mark the checkbox next to the specific service(s) you’d like to monitor. We’ll choose Amazon EC2.

Step 6: Select the time period

Next you’ll want to set the time period that the alarm will be active for. You can choose anything from 10 seconds to an entire day. The interface will also show a graph which plots your current usage in blue against the alarm threshold in red. If the blue line crosses the red at any point during the selected time period, the alarm will activate.

Step 7: Set the threshold

Next, set the threshold over or under which the alarm will ring. We’ll set our alarm to activate if our EC2 charges exceed $10 within the next 6 hours. You can also use the anomaly detection panel to specify a band around your usage outside of which you’d like the alarm to activate.

Step 8: Create a notification

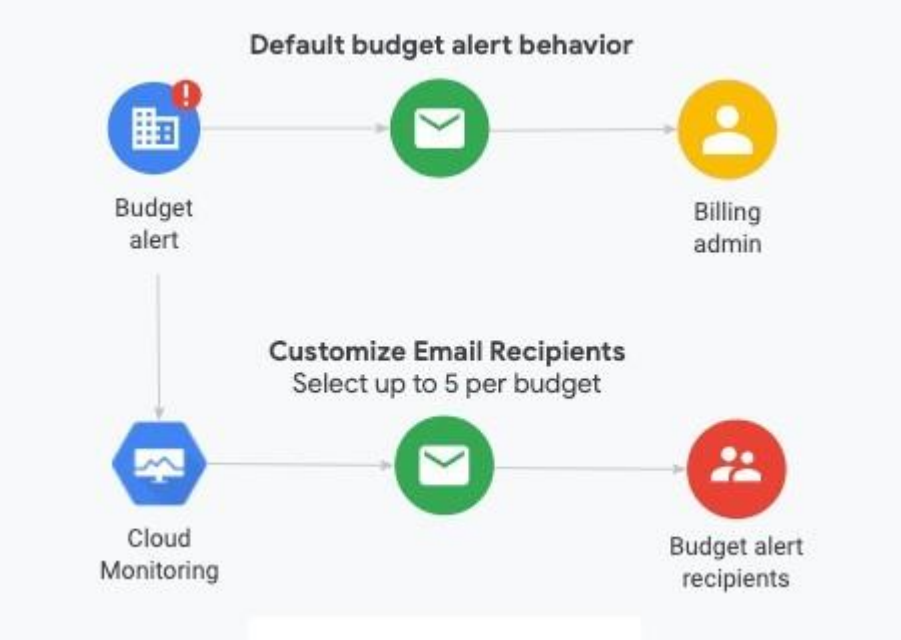
Next you’ll want to create a notification using Amazon SNS. You just click “create new topic”, give it a name, and enter the email you’d like to be notified at into the box.

Step 9: Add a description

Now you can create a name and description that will help you to remember what trigger is set for your alarm.

Step 10: Preview and create

A final page will now open which summarizes and displays your alarm's settings as you've created them. If everything looks good, you can click "Create Alarm" to have your alarm set.



RESULT:

Thus billing alerts for Cloud Organization is created.

EX.NO.5 Compare Cloud cost for a simple web application across AWS, Azure and GCP and suggestion of best one

AIM:

To Compare Cloud cost for a simple web application across AWS, Azure and GCP and suggestion of best one.

PROCEDURE/COMPARISON:

Amazon Web Services

Amazon Web Services is a subsidiary of amazon.com, which provides an on-demand Cloud Computing platform to individuals, companies, and governments on a paid-subscription basis.

Amazon Web Services is the oldest and the most experienced player in the cloud market. As one of the oldest cloud providers, it has established a bigger user base, as well as bigger trust and reliability factors.

Check out Intellipaat's *AWS training* to get ahead in your career!

AWS was publicly launched in 2006 with service offerings such as Elastic Compute Cloud (EC2), Simple Storage Service (Amazon S3), etc. By 2009, Elastic Block Store (EBS) was made public, and services such as Amazon CloudFront, Content delivery network (CDN), and more formally joined the AWS Cloud Computing Service offerings.

Microsoft Azure

Microsoft Azure, initially called Azure, was launched in 2010 with the intent to provide a competent Cloud Computing platform for businesses. Azure was renamed as 'Microsoft Azure' in 2014, though the name 'Azure' is still commonly used. Since its inception, Microsoft Azure has shown great progress among its competitors.

Kickstart your career journey by enrolling in this Google Cloud training in London.

Google Cloud Platform

Google Cloud Platform (GCP), which is offered by Google, is a suite of Cloud Computing services that runs on the same infrastructure that Google uses internally for its end-user products such as Google Search engine, YouTube, and more.

Google Cloud Platform began its journey in 2011, and in less than a decade it has managed to create a good presence in the cloud industry. The initial intent of Google Cloud was to strengthen Google's own products such as Google Search engine and YouTube. But now, they have also introduced their enterprise services so that anyone can use Google Cloud Platform which shares the same infrastructure as that of Google Search or YouTube.

AWS Vs. Azure Vs. Google Cloud: Availability Zones

It has been already established that AWS was the earliest in the cloud domain which means that they have had more time to establish and expand their network. So, AWS is hosting in multiple locations worldwide. Azure and GCP are also hosting in multiple locations worldwide, but the difference occurs in the number of their respective availability zones.

- AWS has 66 availability zones with 12 more on the way.
- Azure has 54 regions worldwide and is available in 140 countries all around the world.
- Google Cloud Platform has been made available in 20 regions around the world with 3 more on their way.

Moving on with this Azure vs AWS vs Google Cloud blog, let's look into the market shares and growth rate of each of these cloud providers.

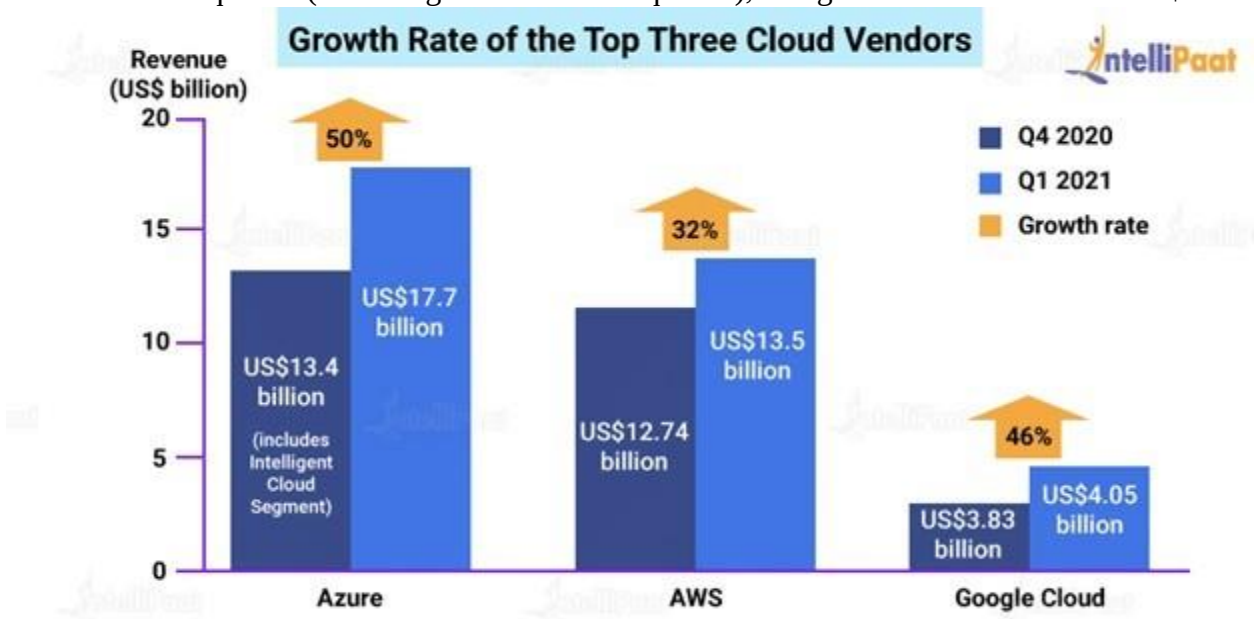
Are you preparing for the AWS interview? Then here are the latest AWS interview questions!

AWS Vs. Azure Vs. Google Cloud: Market Shares and Growth Rate

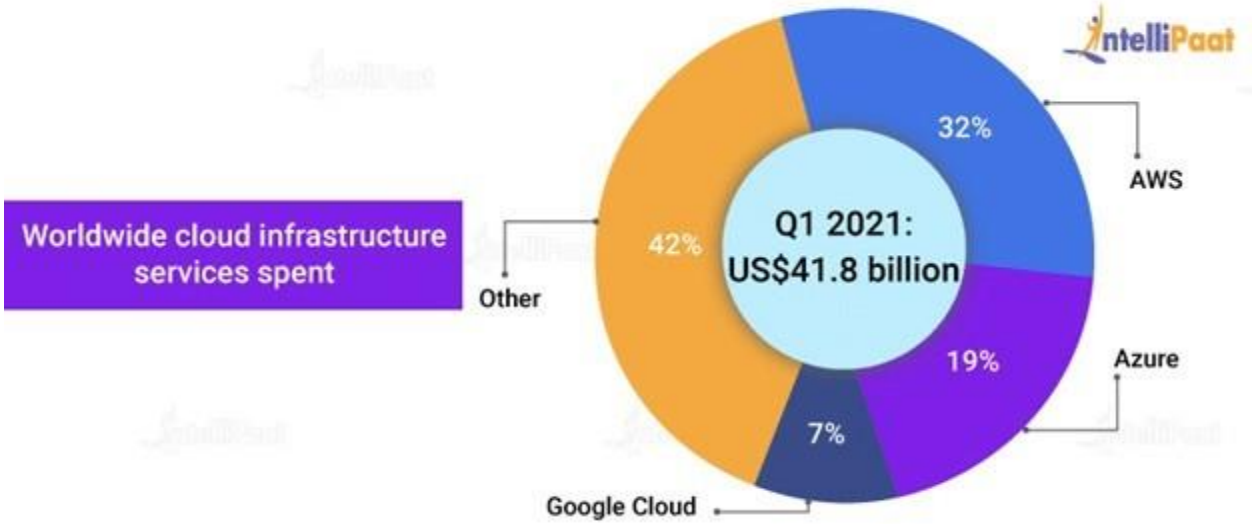
According to the reported quarterly earnings for 2021, Microsoft's Azure cloud revenue has been observed to, once again, outperform both AWS and Google Cloud combined.

In spite of the Goliath-like stature of Amazon's AWS, Microsoft's Azure cloud outperformed its competitors with its US$17.7 billion (50% revenue growth over the previous quarter) in commercial-cloud

revenue as per the fiscal earnings report. While Amazon's AWS reported US$13.5 billion in cloud business revenue for the quarter (revenue grew 32% in the quarter), Google Cloud had a modest US$4.05 billion.



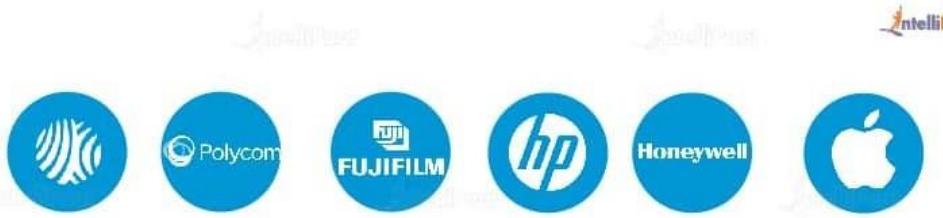
Reports by Canalys mentions that as of April 2021, the global cloud market grew 35% this quarter to $41.8 billion. AWS covers 32% of the market, followed by Azure at 19% and Google at 7%.



AWS Vs. Azure Vs. Google Cloud: Who Uses Them?

Since AWS is the oldest player in the cloud market, it comparatively has a bigger community support and user base. Therefore, AWS has more high-profile and well-known customers like Netflix, Airbnb, Unilever, BMW, Samsung, MI, Zynga, etc.

Azure is also gaining its share of high-profile customers with time. As of now, Azure has almost 80 percent of Fortune 500 companies as its customers. Some of its major customers are Johnson Controls, Polycom, Fujifilm, HP, Honeywell, Apple, etc.



Google, on the other hand, shares the same infrastructure as that of Google Search and YouTube and, as a result, many high-end companies have put their faith in Google Cloud. Major clients of Google Cloud are HSBC, PayPal, 20th Century Fox, Bloomberg, Dominos, and more.

All these cloud providers offer various cloud computing services that are required for any basic business. The difference occurs in the number of these services. So, moving forward with this Azure vs AWS vs Google Cloud blog, let's look into the service offerings of these cloud providers.

AWS Vs. Azure Vs. Google Cloud: Services

With the added advantage of five years of a head start, AWS computing services are by far the most evolved and functionally rich.

Want to read more about AWS and Azure? Go through this *AWS Tutorial* and *Azure Tutorial*!

AWS offers around 200+ services, whereas Azure offers up to 100+ services. Google Cloud, on the other hand, is catching up with Azure and AWS offering around 60+ services.

Service offerings from AWS, Azure, and GCP that come under the domains of compute, database, storage, and networking are mapped below:

Compute Services

Services	AWS	Azure	GCP
IaaS	Amazon Elastic Compute Cloud	Virtual Machines	Google Compute Engine
PaaS	AWS Elastic Beanstalk	App Service and Cloud Services	Google App Engine
Containers	Amazon Elastic Compute Cloud Container Service	Azure Kubernetes Service (AKS)	Google Kubernetes Engine
Serverless Functions	AWS Lambda	Azure Functions	Google Cloud Functions

Database Services

Services	AWS	Azure	GCP
RDBMS	Amazon Relational Database Service	SQL Database	Google Cloud SQL
NoSQL: Key–Value	Amazon DynamoDB	Table Storage	Google Cloud DatastoreGoogle Cloud Bigtable
NoSQL: Indexed	Amazon SimpleDB	Azure Cosmos DB	Google Cloud Datastore

To learn about Google cloud in detail, enroll in this Google Cloud course in United States provided by Intellipaat.

Storage Services

Services	AWS	Azure	GCP
Object Storage	Amazon Simple Storage Service	Blob Storage	Google Cloud Storage
Virtual Server Disks	Amazon Elastic Block Store	Managed Disks	Google Compute Engine Persistent Disks
Cold Storage	Amazon Glacier	Azure Archive Blob Storage	Google Cloud Storage Nearline

File Storage	Amazon Elastic File System	Azure File Storage	ZFS/Avere

Are you preparing for the Azure interview? Then here are the latest *Azure Interview Questions*!

Networking Services

Services	AWS	Azure	GCP
Virtual Network	Amazon Virtual Private Cloud(VPC)	Virtual Networks (VNets)	Virtual Private Cloud
Elastic Load Balancer	Elastic Load Balancer	Azure Load Balancer	Google Cloud Load Balancing
Peering	Direct Connect	ExpressRoute	Google Cloud Interconnect
DNS	Amazon Route 53	Azure DNS	Google Cloud DNS

RESULT:

Thus the Comparison of Cloud cost for a simple web application across AWS, Azure and GCP is studied.

CONTENT BEYOND SYLLABUS

EX:NO:6 Setting up C Programming Environment

Aim

Install a C compiler in the virtual machine created using virtual box and Execute Simple Programs.

Procedure:

Installation of virtual box:

1. Visit http://www.virtualbox.org/wiki/downloads
2. Download Virtual Box platform packages for your OS
3. Open the Installation Package by double clicking.
4. Click continue and finish installing Virtual Box.
5. When finished installation, close the window.

Install Linux in virtual Box:

I. Visit the page http://www.ubuntu.com/download/ubuntu/download
2. Choose the Latest version of Ubuntu and 32-bit and click "Start Download" Click "Continue" on the pop-up window
3. Type VM name, select "Linux" for the OS and Choose "Ubuntu" for the version.
4. Choose create a new virtual hard disk
5. Click Continue or Next
6. Choose VDI (Virtual BoxDisk Image)
7. Click Continue or Next

Click the folder icon and choose the ubuntu is file you downloaded.

8. Select the size of the Virtual Disk (I recommend choosing 8 GB) and click continue
9. Click Create
10. Running Linux:

1. Choose Ubuntu from left column and click Start
2. Click continue on pop-up window
3. Click the folder icon and choose the ubuntu iso file you downloaded and click continue and Start

4. Check “Download updates” and click forward

5. Choose “Erase disk and install Ubuntu” and click Forward (Don’t worry, it won’t wipe your computer)

C Programming on Linux:

1. Open Terminal (Applications-Accessories-Terminal).
2. Open gedit by typing “gedit &” on terminal.
3. Type the following on gedit (or any other text editor)

```
#include & lt; stdio.h&gt;
main()
{
 Printf (&quot; HelloWorld\n& quot;);
 }
```

4. Save this file as “hello world.c”.
5. Type “ls” on Terminal to see all files under current folder.
6. Confirm that “hello world.c” is in the current directory.

If not, type cd DIRECTORY_PATH to go to the directory that has “hello world.c”.

7. Type“gcc helloworld.c” to compile, and type “ls” to confirm that an ew executable file“ a.out” is created.
8. Type“./a. out” on Terminal to run the program.
9. If you see “HelloWorld” on the next line, you just successfully ran yourfirst C program!

RESULT:

Thus the C compiler is installed in the virtual machine created using virtual box and Simple programs executed.