Observability and Management: Configure Alarms with Notifications and Create Monitoring Queries

Lab 20-1 Practices

Get Started

Overview

Oracle Cloud Infrastructure (OCI) Observability and Management provides visibility and actionable insights derived using Machine Learning Algorithms. This platform is open and extensible, and provides cloud-based monitoring and analytics.

Some of the Observability and Management services include Monitoring, Logging, Event Services, Logging Analytics, and Application Performance Monitoring. In this lab, you will create alarms and queries, and trigger alarms.

In this lab, you will:

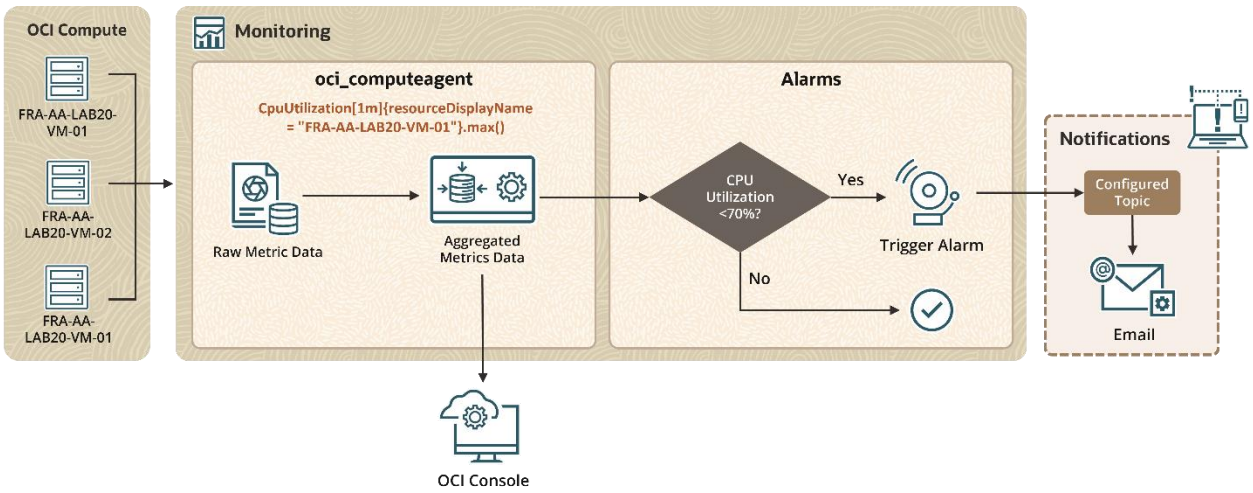
a. Create a Virtual Cloud Network (VCN)

b. Launch three Compute Virtual Machine instances

c. Create alarms and view service metrics

d. Create CPU stress and fire alarms

e. Create queries



Assumptions

- You must be familiar with navigating the OCI Console.
- In this lab, **Germany Central (Frankfurt)** is considered as your region.

Set Up the Environment

In this practice, you will configure the cloud environment, create a virtual network, and compute instances. The resources created in this practice will help you complete the rest of the lab.

Task 1: Create a VCN

A Virtual Cloud Network (VCN) defines a private network in the cloud environment where you can specify networking parameters such as CIDR block and route tables, along with security controls like access control lists and virtual firewalls. You can also allow connectivity to the public Internet. In this task, you will create a VCN.

Note: For a production VCN environment, it is recommended to further restrict network access controls to meet your security requirements.

1. Sign in to your Oracle Cloud Infrastructure (OCI) account.
2. In the console ribbon at the top of the screen, click the Region icon to expand the menu and select **Germany Central (Frankfurt)**.
3. From the navigation menu, under **Networking**, select **Virtual Cloud Networks**.
4. From the left navigation panel, ensure you are in the compartment allotted to you. Click **Create VCN**.
5. In the **Create a Virtual Cloud Network** dialog box, populate the following information:
    - **Name**: FRA-AA-LAB20-1-VCN-01
    - **Create In Compartment**: <*your compartment*>.
    - **IPv4 CIDR Block**: 10.0.0.0/16 (Press **Enter** to add the IP block.)
6. Leave other fields as default. Click **Create VCN**.
7. After the VCN is created, click **FRA-AA-LAB20-1-VCN-01** VCN to view the details page. Under **Resources** in the left navigation panel, click **Internet Gateways**.
8. Click **Create Internet Gateway**.

9. In the **Create Internet Gateway** dialog box, populate the following information:
    - **Name**: FRA-AA-LAB20-1-IG-01
    - **Create In Compartment**: <*your compartment*>
10. Click **Create Internet Gateway**.
11. Next, make a quick update to the VCN route table to make use of the Internet Gateway created in the previous step. Under **Resources** in the left navigation panel, click **Route Tables**.
12. Click **Default Route Table for FRA-AA-LAB20-1-VCN-01** and then, click **Add Route Rules**.
13. In the **Add Route Rules** dialog box, populate the following information:
    - **Target Type:** Internet Gateway
    - **Destination CIDR Block**: 0.0.0.0/0
    - **Target Internet Gateway**: FRA-AA-LAB20-1-IG-01
14. Click **Add Route Rules** to complete the process.
15. Finally, create a subnet in the VCN to identify IP space and deploy a VM. Return to the VCN details page by clicking **FRA-AA-LAB20-1-VCN-01** in the breadcrumb list at the top of the page.
16. Under **Resources** in the left navigation panel, click **Subnets**. Then, click **Create Subnet**.
17. In the Create Subnet dialog box, populate the following information:
    - **Name**: FRA-AA-LAB20-1-SNET-01
    - **Create In Compartment**: <*your compartment*>.
    - **Subnet Type**: Regional (Recommended)
    - **IPv6 CIDR Block**: 10.0.0.0/24
    - **Route Table Compartment in <*your compartment*>**: Default Route Table
    - **Subnet Access**: Public Subnet
18. Leave other fields as default. Click **Create Subnet**.

Task 2: Set Up SSH Keys for Virtual Machine Instance

Before launching a Virtual Machine instance, you will create SSH keys to authenticate the Instance using Oracle Cloud Shell.

1. In the OCI Console ribbon at the top of the screen, ensure that the correct Region is selected. In this case, the region is **Germany Central (Frankfurt)**.

2. Click **Cloud Shell** icon next to the region.

3. In the Cloud Shell, ensure that you are in the home directory of your account. To check, run the following command:

   ```
   $ pwd
   ```

   Reminder: Do not include the `$` symbol when pasting code into Cloud Shell.

   If you are in your home directory, the value will be `/home/<user_name>`.

4. To change the directory to `.ssh` directory, run the following command:

   ```
   $ cd .ssh/
   ```

5. If the previous step shows an error as "No such file or directory," then run the following command:

   ```
   $ mkdir .ssh/
   ```

6. Now, change directory to `.ssh/` by running the following command:

   ```
   $ cd .ssh/
   ```

7. To create ssh keys, run the following command:

   ```
   $ ssh-keygen -b 2048 -t rsa -f sshkeys
   ```

8. Do not enter a password when prompted, press **Enter**.

   Note: There are two files saved into the `.ssh` directory: **sshkeys.pub** (public key) and **sshkeys** (private key). **sshkeys.pub** will be used while creating compute instances, and **sshkeys** will be used to authenticate.

9. Run the following command to view the contents of the **sshkeys.pub** public key.

```
$ cat /home/<user_name>/.ssh/sshkeys.pub
```

Note: Replace <user_name> with your username as noted in step 3.

10. Copy and paste the content of **sshkeys.pub** public key into a Notepad file. You will use this content while creating compute instance.

11. Close the Cloud Shell by clicking **X** at the top-right corner. Then, click **Exit**.

Task 3: Launch Compute Virtual Machine Instance

Now, you will launch a Virtual Machine in your newly created VCN. For this lab, you will create three instances.

1. In the OCI Console ribbon at the top of the screen, ensure that you have selected the same region where you created the VCN.

2. From the navigation menu, under **Compute**, click **Instances**.

3. From the left navigation panel, ensure that you are in the compartment allotted to you. To create the first instance, click **Create instance**.

4. In the **Create compute instance** dialog box, enter **FRA-AA-LAB20-1-VM-01** in the **Name** field.

5. In the **Create in compartment** field, select <*your compartment*>.

6. The **Availability Domain** will be pre-populated to match the subnet you created earlier.

7. Ensure that the **Image** is selected as **Oracle Linux 8**. If not, click **Change Image** and select **Oracle Linux 8**.

8. In the **Shape** field, click **Change Shape**. Then select **VM.Standard.A1.Flex** (1 OCPU, 6GB Memory) [Shape series: Ampere].

   Note: Your options and naming conventions may not match exactly as given here, so select an appropriate shape if it is shown different for your region.

9. In the **Primary network** field, select **Select Existing Virtual Cloud Network** and ensure **FRA-AA-LAB20-1-VCN-01** is specified in the **Virtual cloud network** field.

10. In the **Subnet** field, select **Select Existing Subnet**. Ensure the **Subnet** is specified as **FRA-AA-LAB20-1-SNET-01**.

    If not, double-check the compartment is set to <*your compartment*>. You may have to switch to a different Availability Domain (see above – the Availability Domain of your subnet and compute instance must match) to allow the selection of your existing subnet, if not already selected.

11. In the **Public IP address** field, select **Assign a public IPv4 address**.

12. In the **Add SSH keys** field, select **Paste public keys**. Then copy the sshkeys.pub public key from the Notepad (copied earlier in previous task) and paste it in the **SSH keys** field.

13. Keep the other options default and click **Create**. The first compute instance is successfully created.

14. Navigate back to the **Instances** page from the navigation menu. Ensure that the **State** of the instance you just created is **Running**.

15. Copy the Public IP corresponding to the **FRA-AA-LAB20-1-VM-01** instance and paste it in the Notepad.

16. Now, click the **Cloud Shell** icon next to the Region at the top of the screen.

17. Run the following command with pasting the sshkeys - private key and Public IP:

    ```
    $ ssh -i /home/<user_name>/.ssh/sshkeys opc@X.X.X.X
    ```

    - Replace `<user_name>` with your username.
    - Replace *X.X.X.X* with the public IP address copied in step 15.

    Note: The SSH Key is the private key created in the previous task. It is used to authenticate.

18. Enter **Yes** when prompted to connect and ensure you are connected to the instance.

19. Enter **exit** to close the connection.

20. To create a second instance, repeat steps 2 through 7. Keep all settings the same except the **Name** of the instance. Enter the **Name** of the second instance as **FRA-AA-LAB20-1-VM-02**.

21. In the **Shape** field, click **Change Shape**. Then select **VM.Standard.A1.Flex** (1 OCPU, 6GB Memory) [Shape series: Ampere].

22. In the **Public IP address** field, select **Do not assign a public IPv4 address**. In the **Add SSH keys** field, select **No SSH keys**.

    Note: The instance is not required to be accessed; therefore, assigning a Public IP address and SSH keys for this instance can be skipped.

23. Keep the other options default and click **Create**. The second compute instance is successfully created.

24. Navigate back to the **Instances** page from the navigation menu. Ensure that the State of the second instance created is **Running**.

25. To create a third instance, repeat steps 2 through 7. Keep all settings the same except the **Name** of the instance. Enter the **Name** of the second instance as **`FRA-AA-LAB20-1-VM-03`**.

26. In the **Shape** field, click **Change Shape**. Then select **VM.Standard.A1.Flex** (1 OCPU, 6GB Memory) [Shape series: Ampere].

27. In the **Public IP address** field, select **Do not assign a public IPv4 address**. In the **Add SSH keys** field, select **No SSH keys**.

    Note: The instance is not required to be accessed; therefore, assigning a Public IP address and SSH keys for this instance can be skipped.

28. Click **Create**. The third compute instance is successfully created.

29. Navigate back to the **Instances** page from the navigation menu. Ensure that the State of the third instance created is **Running**.

Create Alarms and View Service Metrics

In this practice, you will view the service metrics for your instances, confirm that the required monitoring plug-in is enabled, and set up alarm notifications.

Task 1: Confirm that Compute Instance Monitoring Plug-In Is Enabled

To view the service metrics available in the OCI Console, the compute instance monitoring plug-in must be enabled. This plug-in emits metrics about the instance's health, capacity, and performance—such as CPU and memory utilization.

Note: The plug-in will be enabled by default, but it should be confirmed.

1. From the OCI Console navigation menu, under **Compute**, select **Instances**.
2. Click the instance **FRA-AA-LAB20-1-VM-01**.
3. Click **Oracle Cloud Agent** tab.
4. Scroll down to find the **Compute Instance Monitoring** plug-in and ensure that it is running and enabled.
5. Navigate back to the **Instances** page and repeat steps 1-4 for the instance **FRA-AA-LAB20-1-VM-02**.
6. Repeat steps 1-4 for the instance **FRA-AA-LAB20-1-VM-03**.

Task 2: Create a Topic and a Subscription Inside a Topic

Now that you have confirmed that Monitoring is enabled, you will create an alarm that is triggered when the service metrics reach a designated threshold. You will see this alarm gets triggered later in the practice when you perform a CPU stress test.

To create an alarm, you must first create a notification so that the alarm has a way to notify the relevant parties. For example, an alarm can email an administrator when a CPU usage threshold has been breached.

1. From the OCI Console navigation menu, select **Developer Services**. Under **Application Integration**, select **Notifications**.
2. From the left navigation panel, ensure you are in the compartment assigned to you.

3. Click **Create Topic**.

4. In the **Create Topic** dialog box, enter **FRA-AA-LAB20-1-TOP-01** in the **Name** field and enter **Description** if required as its optional.

5. Click **Create**.

6. Once the topic state changes to **Active**, click the topic to view the details.

7. Under **Resources**, click **Create Subscription**.

8. In the **Create Subscription** dialog box, select **Email** in the **Protocol** field.

9. In the **Email** field, enter your email address.

10. Click **Create**.

11. Click the subscription that you just created.

12. The Subscription Information will be displayed with the status as **Pending Confirmation**.

13. Check the email account you specified and click the "Confirm subscription" verification link in it. A pop-up browser window will tell you that the subscription has been confirmed.

14. Navigate back to the **Subscriptions** page and verify that the subscription status has changed to **Active**.

    Note: You may need to refresh your browser if the status is not updated.

A topic and a subscription inside a topic are successfully created.

Task 3: Create an Alarm for CPU Utilization

Now that you've created the topic and subscription for a notification, you will create your alarm. This alarm will be activated when the CPU utilization reaches a threshold that you designate.

1. From the OCI Console navigation menu, select **Observability & Management**. Under **Monitoring**, click **Alarm Definitions**.

2. From the left navigation panel, ensure that you are in the compartment assigned to you.

3. Click **Create Alarm**.

4. In the **Create Alarm** dialog box, populate the following information in the **Create alarm** section:

    - **Alarm name**: FRA-AA-LAB20-1-ALA-01
    - **Alarm severity**: Critical
    - **Alarm body**: High Usage of CPU

5. The **Tags** section is optional. Therefore, keep the default selections.

6. Populate the following information in the **Metric description** section:

    - **Compartment**: <*your compartment*>
    - **Metric namespace**: oci_computeagent
    - **Metric name**: CpuUtilization
    - **Interval**: 1m
    - **Statistic**: Max

    Note: The **Resource Group** field is optional. Therefore, you can skip it for now.

7. Populate the following information in the **Metric dimensions** section:

    - **Dimension name**: resourceDisplayName
    - **Dimension value**: FRA-AA-LAB20-1-VM-01

8. Populate the following information in the **Trigger rule** section:

    - **operator**: greater than
    - **Value**: 70
    - **Trigger delay minutes**: 1

9. Populate the following information in the **Define alarm notifications** section:

    - **Destination service**: Notifications
    - **Compartment**: <*your compartment*>
    - **Topic**: FRA-AA-LAB20-1-TOP-01

You have created the topic earlier and recall that the topic is the communication channel, such as email. When the alarm is triggered, a notification is sent to the subscribed email addresses.

10. Select the option **Split notifications per metric stream** in the **Message grouping** section.

    With this setting, you are configuring the Alarm to send a message for the specific instance when it reaches the CPU threshold. The UI shows a message which is just a reference- **Consider limits when the alarm contains a high number of metric streams.**

11. You can select the message format, which is generally the first option, **Send formatted messages**.

12. You can also choose to have a notification repeated at certain frequencies if an alarm continues. Keep the **Repeat notification** option deselected.

13. You have the option to suppress the notification. Keep the **Suppress notifications** option deselected.

14. Select **Enable this alarm** and click **Save Alarm**.

You should now be able to see the alarm's details.

Create CPU Stress and Fire Alarm

In this practice, you will create a CPU Stress on the first instance (FRA-AA-LAB20-1-VM-01), monitor the effect of CPU stress on the instance, and see an event triggered when the CPU utilization is greater than the threshold, which causes the alarm to fire.

Task 1: Create CPU Stress for an Instance

Now that you have created an alarm, Observability and Management monitors the working of instances and sends a notification when the alarm is triggered. For this purpose, the CPU is subjected to stress and forced to run to its maximum capacity. When the CPU Utilization metric is greater than the threshold value, the alarm gets triggered.

This is simulated by means of a CPUStress generator. The following steps are with respect to a Linux OS.

1. From the OCI Console navigation menu, under **Compute**, click **Instances.**
2. Click the instance **FRA-AA-LAB20-1-VM-01.** Copy the Public IP address.
3. Click the **Cloud Shell** icon from the Console ribbon at the top of the page.
4. Connect to the instance by running the following command:

   ```
   $ ssh -i /home/<user_name>/.ssh/sshkeys opc@<X.X.X.X>
   ```

   - Replace `<user_name>` with your username.
   - Replace *X.X.X.X* with the public IP address.
5. You should get a message that the FIPS mode is initialized.
6. Run the following command to install the EPEL (Extra Packages for Enterprise Linux) repository on Linux distributions to install additional standard open-source software packages by using YUM and DNF package manager. If you are asked if it is OK, enter **Y**.

   ```
   $ sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
   ```
7. Enter **Y**. You will see **Complete!** when it is complete.

8. Install the stress package. Stress is a generator tool, devised to subject your system to configurable measure of CPU, memory, I/O, disk stress. To install, run the following command:

   ```
   $ sudo yum install stress
   ```

   Note: If you are asked if it is OK, enter **`y`** again.

You will get a message when the installation is successful.

Task 2: Include Stress to the Compute Instance

Now, you need to induce stress to the instance. The stress on the compute instances increases on repeated use of the stress command. Run the following command:

```
$ uptime
$ stress --cpu 8 --timeout 300
```

Task 3: Trigger the Alarm

1. From the OCI Console navigation menu, select **Observability & Management**. Under **Monitoring**, click **Alarm Definitions**.
2. Click **FRA-AA-LAB20-1-ALA-01** alarm that you created earlier.
3. The icon in FRA-AA-LAB20-1-ALA-01 would have changed to Firing mode due to the stress induced. This happens when the load on the CPU Utilization crosses the threshold limits. Please wait for a minute if the status is not changed to Firing, and then refresh the page.
4. Scroll down to the **Alarm history** graph, which signifies that the CPU stress has surpassed the set threshold.
5. An email notification is sent to the configured subscription email of the Notifications Topic as Alarm status changes from OK to Firing.
6. The email provides details about Alarm OCID, Number of Metrics breaching threshold, and Dimensions.
7. Navigate back to the **Alarm Definitions** page and select the check box against the FRA-AA-LAB20-1-ALA-01 alarm.

8. Click **Actions** and select **Add suppressions** from the drop-down list.

9. In the **Suppress Alarms** Wizard, select the default **Start time** and **End time** and click **Apply suppressions** to confirm.

10. Click **Close** and verify that the column **Suppressed** shows the alarm is suppressed for the period.

11. Click the **Cloud Shell** icon to open Cloud Shell where the stress was initiated on the Instance. Press Ctrl + C to stop the stress.

12. Navigate back to the **Alarm Definitions** page and click the **FRA-AA-LAB20-1-ALA-01** alarm.

13. The CPU-usage-alarm icon would have changed to OK mode as the stress is now stopped.

14. Verify an email notification is not received by the configured subscription email for the status being changed from Firing to OK. This notification is not sent due to Alarm being suppressed for the period.

Create Queries

In this practice, you will create different types of queries and see how they are all represented graphically.

Task 1: Create Standard Queries

In this task, you will learn about query expressions and components, and you will execute sample queries that can be used with the Monitoring service. The Metrics Explorer creates queries that are used to search and aggregate metric data points collected from resources.

A standard query includes a metric namespace (the source or application being measured), metric (what is being measured), interval (over what period), and statistic (how it's being measured, e.g., a sum, rate, or max value).

1. From the OCI Console navigation menu, select **Observability & Management**. Under **Monitoring**, click **Metrics Explorer**.
2. To create a standard query, populate the following information in the **Query** section:
    - **Compartment**: <*your compartment*>
    - **Metric namespace**: oci-computeagent
    - **Metric name**: CpuUtilization
    - **Interval**: 5m
    - **Statistic**: Max
3. Click **Update Chart**.

The chart generated is the output of the query. It represents the CPU utilization (CpuUtilization) of all instances (oci_computeagent) in the past five minutes. The corresponding Monitoring Query Language (MQL) is displayed under Query 1.

Task 2: Create Standard Queries with a Filter

A filter condition is used along with a standard query to display the graphs that satisfy specific conditions. The filter condition is entered in the Metric Dimensions area and includes a name and (optional) a value.

1. From the navigation menu, select **Observability & Management**. Under **Monitoring**, click **Metrics Explorer**.
2. Populate the following information to create a grouping function using Basic mode in the **Query** section:
    - **Compartment**: <*your compartment*>
    - **Metric namespace**: oci-computeagent
    - **Metric name**: CpuUtilization
    - **Interval**: 5m
    - **Statistic**: Max
3. In the **Metric dimensions** section, populate the following information:
    - **Dimension name**: availabilityDomain
    - **Dimension value**: *Select an availability domain.*
4. Click **Update Chart**.

The chart displays the CPU utilization of the compute instances in an interval of five minute for the inputted availability domain.

Task 3: Create Aggregation Using Basic Queries

Simple aggregation (grouping) function queries return the combined value of all metric streams for the selected statistic. They can be written manually in the Query Code Editor pane by checking the Advanced mode option, or you can use the Standard Query mode used above.

1. From the navigation menu, select **Observability & Management**. Under **Monitoring**, click **Metrics Explorer**.

2. Populate the following information to create a grouping function using Basic mode in the **Query** section:

    - **Compartment**: <*your compartment*>
    - **Metric namespace**: oci-computeagent
    - **Metric name**: CpuUtilization
    - **Interval**: 5m
    - **Statistic**: Max

3. In the **Metric dimensions** section, populate the following information:

    - **Dimension name**: availabilityDomain
    - Select the **Aggregate metric streams** check box.

    Note: You can leave the **Dimension value** field blank for now.

4. Click **Update Chart**.

The graph displays the aggregation of CPU utilization of all availability domains, with an interval of five minutes, and a statistic option of the Max function.

The selection of **Aggregate metric streams** check box is referred to as **grouping** function while using Advanced mode. This query can be viewed with selecting **Advanced mode** check box.

Task 4: Create Advanced Queries

The nested queries are written as part of the Advanced mode in the **Query code editor**.

1. From the navigation menu, select **Observability & Management**. Under **Monitoring**, click **Metrics Explorer**.

2. Select the **Advanced mode** check box at the top right of the **Query 1** section.

3. Populate the following information to create a grouping function using Basic mode in the **Query** section:

   - **Compartment**: <*your compartment*>
   - **Metric namespace**: oci-computeagent

4. Enter the following code in the **Query code editor** field.

   ```
   (CpuUtilization[1m].max() > 5).grouping().max()
   ```

5. Click **Update Chart**.

   The displayed output groups the compute instances and displays the ones whose CpuUtilization is more than 5 percent in the past minute.

GroupBy is a grouping function, which can be written using Advanced mode. It is another way to aggregate metric streams. For example, you can group by **shape** used by the Instance.

1. To group by shape, enter the following code into the **Query code editor**.

   ```
   CpuUtilization[5m].groupBy(shape).max()
   ```

2. Click **Update Chart**.

   The displayed output groups compute instances by shape and displays the CpuUtilization with an interval of 5 mins and showing the maximum reported value in the graph.