



# IJIRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

e-ISSN: 2319-8753 | p-ISSN: 2320-6710



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 7, July 2022

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

Impact Factor: 8.118



9940 572 462



6381 907 438



ijirset@gmail.com



www.ijirset.com

# Modern Pragmatic Cloud-based Exception Alerts and App Availability Notifications

Sohail Ansari<sup>1</sup>, Nitin Sahu<sup>1</sup>, Dr. Shivkumar Goel<sup>2</sup>

Student, Department of MCA, Vivekanand Education Society's Institute of Technology, Collector Colony, Chembur, Mumbai, India<sup>1</sup>

Associate Professor/HOD, Department of MCA, Vivekanand Education Society's Institute of Technology, Collector Colony, Chembur, Mumbai, India<sup>2</sup>

The objective of this service is to use the power of Azure cloud and enhance legacy services. Using the potential of Azure, an alert be raised whenever an exception is generated in the Application insights or when the Web app service goes down or is back online and trigger a logic app to send an email about it.

**ABSTRACT:** The most important aspect of any production deployed application is to be bugfree and available for majority of the time, and both the requirements currently in the market has either a manual high labor or legacy solution. Exception management and service availability also help with many SLA's that are signed which suggests that the service is available almost all the time and if not, an alert will notify about its down time, and whenever an exception is generated the right team and people can investigate it and rectify it as soon as possible.

To counter this we can use the power of cloud computing which can do it for us. Using application insights, we can log all the exception being generated by the application and using service health in azure we can generate alerts whenever the application goes down or comes back online; and eventually using azure logic apps to notify the clients and dev teams about the concerning issue.

**KEYWORDS:** Application Insights, Azure Logic Apps, Service Setup, Logic app.

## I. INTRODUCTION

Application Insights Overview: Application insight is characteristics of Azure Monitor that provide extensible APM i.e. application performance management and live web apps are monitored. The Programmers and cloud professionals can use application insights to:

1. To detect performance anomalies automatically.
2. Use advance analytics tools to help solve issues.
3. To see what apps can provide to users.
4. To improve continuously performance of app and usability

Application Insights:

1. It has many platforms like .Net, Node.js, Java and many more.
2. It works for any apps whether is deployed on-premises, hybrid, or public cloud.
3. Helps in DevOps processes for integration.
4. Many development tools are connected with connection points.
5. Using mobile apps by integrating with visual studio App center the telemetry can be monitored and analyzed.

Azure Logic Apps

Azure logic apps integrate our apps, data, services, and systems in a cloud-based platform by creating and running automated workflows. For many enterprises and business-to-business (B2B) we can develop as soon as possible highly

scalable integration solutions. A member of Azure Integration Services we can simplify hybrid environments, modern and cutting edge systems across cloud, the way we connect legacy by using azure logic apps.

These following can be automated using Azure Logic Apps service:

1. We can send emails notification using Office 365 if any event occurs.
2. Order process of customer throughout on-premises systems, cloud services and route.
3. Can move uploaded files to Azure Storage From an FTP or SFTP.
4. the task or alerts for items that has to review is created after analyzing and monitoring tweets.

Alerts in Microsoft Azure

In Azure Monitor it alerts its customer proactively by monitoring their data if any issue occurs in customer's application or infrastructure. So, we can solve the system issue before it is noticed by the users of the systems.

Action after alert fires and alerts from these alerts rules are separated. The alert rule decides the criteria and target for alert. State can be disabled or enabled by alter rules. Alerts can never be fired when it is disabled.

Key points of alert rules:

Target Resource – for alerting signal and scope is defines by this. Any of the resource of Azure can be target. Example targets:

- Storage accounts.
- Virtual machines.
- Application Insights.
- Log Analytics workspace.

For certain resources (like virtual machines), a target can be specified multiple resources of the alert rules.

Signal - target resource emits. Some of the types of signal: metric, activity log, Application Insights, and log.

Criteria – A target which is combination of signal and logic. Examples:

- 70 % < Percentage PCU
- 4 ms< Server Response Time
- 100 < Result count of a log query

Alert Name - A specific name which is configured by the user of alert rule.

Alert Description - A Description which is configured by the user of alert rule.

Severity - The severity of the alert after the criteria specified in the alert rule is met. Severity can range from zero to four.

## II. LITERATURE SURVEY

Currently in the market there are no single solution to manage application availability and exception alerts together.

For an organization to handle exceptions it must have a team of support executives who investigate problems when they occur and another support team to handle client support cases if the application goes down.

This decreases the productivity and increases human dependency and a delayed and not necessarily appropriate response from the designated division.

### III. METHODOLOGY

The legacy approach for exception management was to store everything in a database table and whenever there is any problem faced by any client, we go back to the database to investigate the issue. This creates a problem that the solution provided is reactive and not proactive. We are providing solutions to problem when they occur, rather we should be proactively involved such that we can even fix the issue before it is reported.

### IV. SERVICE SETUP

Health check configuration in web app resource for service availability

We need a health endpoint which returns a 200 OK response which will be constantly hit by the web app to check if the service is healthy.

1. To achieve this, we will have a controller which does the same. We will check all the services used in the app to check if everything works fine.  
If yes, we get a 200 OK response; if not, we get 500 Internal service error, and the web app will be shown as unhealthy.
2. When we go on azure devOps in that the app service resource overview tab, the health check config will be displayed as Not Configured, click it.
3. We will get a screen, add the endpoint URL created 1<sup>st</sup> step in Path Textbox.
4. After a few minutes, if the web's health check config shows 100% healthy, the check is setup & the process is completed.  
If it shows degraded, then check if the web app is Active, the endpoint is published and other general things that might give the health check a response other than 200 OK.

Logic app IAM visibility configuration for channel alerts

Pre-requisite: Configure any logic app according to your requirement in your subscription of choice, but the resource group should match the resource group of the existing application insight.

1. Head to the logic app -> Identity -> turn the system assigned status to On.
2. Go the web app where you want to setup alerts -> Access control (IAM) -> Add -> Add role assignment -> Select Reader access -> Next -> Managed identity -> Select members -> Search for the logic app in the appropriate subscription and select and complete the process.
3. The Logic app now has read access to the web app.

Service Availability alerts.

Pre-requisite: Application insights is configured for the web app resource.

Adding action group for alerts

1. To send alerts to anyone, we need an alert group which can send emails / SMS / notification to a group of people.  
For this, go to the particular app service (web app) resource -> Alerts (in the monitoring menu) -> Create alert rule.
2. A blade will be opened, asking for subscription & group details.
3. For the next step we need an email ID which will notify us about the alert. We needed the email in a teams channel so we generated it by going the particular channel -> 3 dot context menu -> Get email address and copy.
4. Add the details shown in screen in 1<sup>st</sup> step, update the email to the id where alert email is to be received and click OK and next.
5. Select the logic app discussed in the previous section in action step. This logic app will be triggered when the alert is fired.
6. Review all the changes and create the action group.

Creating service availability alert rule:

1. Go to the app service (web app) resource -> Alerts (in the monitoring menu) -> Create alert rule .
2. Search for health check and add the condition .
3. To configure the signal logic, change alert logic to configurations and click done.
4. Next step is to add the previously created action group which send an alert if the criteria set in figure previous step is met; Click on Add action group & select the appropriate group and click next.
5. Add the required alert rule details in this step.
6. Review and save the changes.

Creating exception alert rule

Head to the configured application insights -> Alerts -> Create -> Alert rule.

1. Add conditions -> search for exceptions and select it
2. Configure the condition set aggregation granularity(period) 1 hour and frequency of evolution 30 minutes.  
This will ensure that whenever at least 1 exception has occurred over a 30-minute period, this alert will be triggered.
3. Click on add action group -> select the action group created at the start of this section.
4. Now Add the alert rule details in this step.
5. Click on Create alert group and this alert will also trigger the logic app to send the exception email in teams channel.

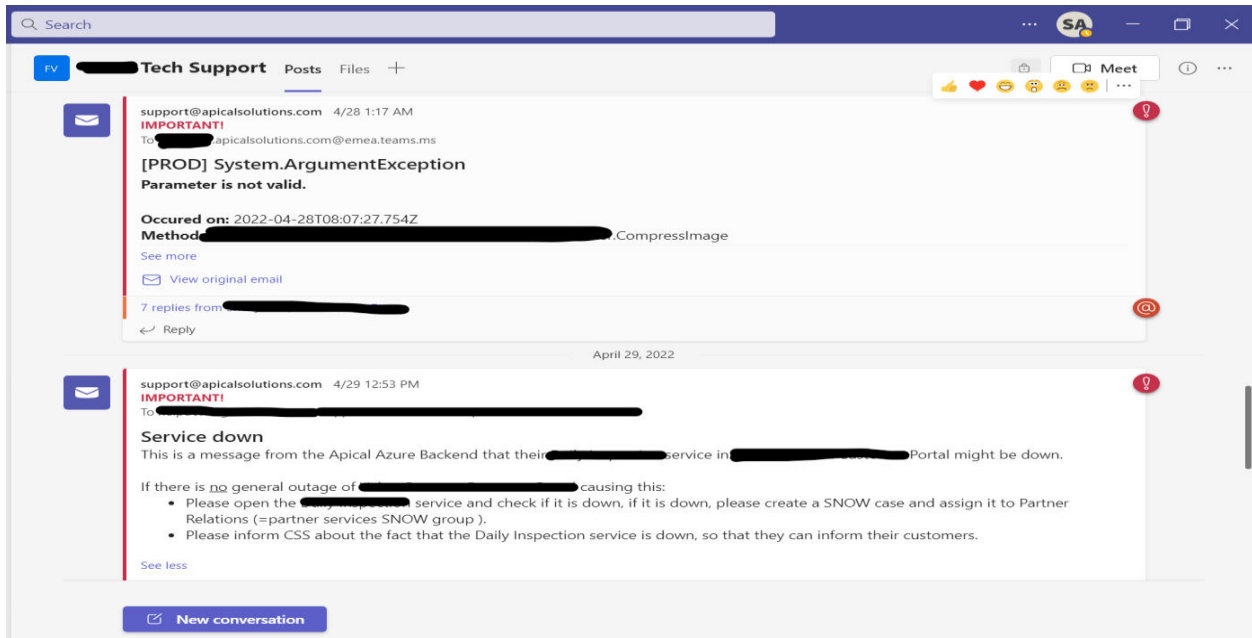
Logic app flow

Pre-requisite: Follow this to obtain App Id and App key to connect to the Application insights connector





## V. RESULT



## VI. CONCLUSION

if there is a larger client base, it is essential that the application remains functional at mostly all times. If for some known or unknown reason the application goes down, the client network will start reporting the issues. We can mitigate this by integrating a health check so that the application can be pinged in frequent intervals to check if the application is function, if there is any problem, it should be notified to the developers and the clients proactively.

Combining both creates an ecosystem such that an organization can give support proactively and in an automated manner with negligible human interference.

The solution provided will work with any tech stack just the implementation might differ for each.

## REFERENCES

- [1]. <https://docs.microsoft.com/en-us/azure/logic-apps/>
- [2]. <https://docs.microsoft.com/en-us/azure/azure-monitor/app/app-insights-overview>
- [3]. <https://docs.microsoft.com/en-us/learn/paths/az-900-describe-cloud-concepts/>
- [4]. <https://docs.microsoft.com/en-us/azure/azure-monitor/alerts/alerts-overview>



**INNO SPACE**  
SJIF Scientific Journal Impact Factor  
**Impact Factor: 8.118**



**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
**INDIA**



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



**www.ijirset.com**

Scan to save the contact details