Initial rough observations:

Combine the cloud watch alarams – composite alarams –

<https://aws.amazon.com/about-aws/whats-new/2020/03/amazon-cloudwatch-now-allows-you-to-combine-multiple-alarms/?GAdoption_CWCompositeAlarmsBlog&sc_icampaign=Adoption_Campaign_pac-edm-2020-CloudWatchConsole_CompositeAlarms&sc_ichannel=ha&sc_icontent=awssm-4982_all_users&sc_ioutcome=Enterprise_Digital_Marketingpac-edm-2020_M&sc_iplace=console-cw&trk=ha_a134p000003yNlhAAE~ha_awssm-4982_all_users&trkCampaign=pac-edm-2020_M>

Create business/use case alerts

[TargetTracking-table/MemberActivityRecords/index/contactDate-index-AlarmLow-791dd965-ed2b-4bdf-9b3f-9171ad7d2fdd](https://us-east-2.console.aws.amazon.com/cloudwatch/home?region=us-east-2#alarmsV2:alarm/TargetTracking-table$2FMemberActivityRecords$2Findex$2FcontactDate-index-AlarmLow-791dd965-ed2b-4bdf-9b3f-9171ad7d2fdd!~(alarmStateFilter~'ALARM)) – reading this name makes no sense

Differentiate between warnings and alerts

Clear segregation of notification types (info warnings and alerts)

Early identification of weather if this is impacting the customer or just the backend process

Creating a page for leadership/business users to show the health of the system and showing historical success/good health

Do a POC on Devops guru for application availability, for business functionality we can use ELK

Customer interaction traction – what did he do in last 15 minutes

Consolidate the service map traces

We have data in CloudWatch service map traces xrays. This needs to be consolidated into a readable and meaningful dashboard

We should be able to track the triaging and maintain the knowledge so far to debug the defect

Hyper care to steady state

Alert and monitoring refinement plan – every feature leading to steady state is in progress

We need a plan to refine our alert and monitoring process

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Observations:**

Alarms: (I have checked the Alarams there are composite laramas configured already)

1. Currently we have a lot of alarms configured (256) leading to a lot of alarm noise. The notifications coming from these alarms are not properly segregated. Managing huge number of alarms is very time consuming.

Logging and debugging:

1. Currently we are not able to perform customer interaction traction (getting the full history of customers operations in the previous 15 minutes)
2. Need to investigate excel mapping sheet to figure out which log groups should we query in CloudWatch to debug the issue
3. X-rays data is not consolidated

**Recommendations:**

Alarms

1. We can use CloudWatch composite alarms to combine multiple alarms together and reduce alarm noise. We can easily combine alarms into alarm hierarchies that only trigger once when multiple alarms are fired helping us to focus on critical operation issues and smaller number of alarm signals
2. Clear segregation of alarms like informational warning and alerts can help work more efficiently
3. Configure business/use-case base alerts

Logging and debugging

1. By consolidating the data from logs X-ray traces and service maps we should provide a way in which just with the help of traceId we should be able to see what the activity is for the past 15 minutes for an individual user. This will help with better triaging of the defects and considerably decrease the defect triaging time. We can also identify any functional errors in an early
2. Create a cloudwatch dashboard activity dashoboard – manoj elk 3rd party
3. We should have a tool or framework in place which automatically maps the error messages to the respective log groups and provide it along with notification or incident
4. Data from Xrays, traces and cloudwatch needs to be consolidated into a readable and meaningful dashboard

General

1. Create a dashboard/page for leadership/business users to show the health of the system and showing historical success/good health
2. Have a system or framework that can track the triaging and maintain the knowledge so far to debug the defect

**Thoughts in progress**

1. Create a POC on DevopsGuru to establish how it can improve the application performance and availability
2. Enable Anomaly Detection to CloudWatch metrics to create a model of metrics expected values by using CloudWatch Machine Learning algorithms
3. Add a new dimension to defects telling whether it is a performance issue or wrong data
4. Synthetic monitoring and blue-green deployment

-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Actions:

* + - 1. Alarms p2-p3 proactive approach (Suggestion by Sekhar)