TrackIt

Whitepaper

Anomaly detection

Is There A Leak in Your Ship?

Table of Contents

The Need to Identify Unwanted Expenses on AWS

The complexity of AWS, unfortunately, results in the inability for most users to accurately monitor and track their AWS cloud costs. The lack of an effective cloud cost monitoring system often leads to users incurring increments in their costs with no means to identify such unnecessary expenditure.

$16,000 in Unexpected AWS Expenses in 2 Months

Keypr, one of TrackIt's client companies was nothing short of surprised when they realized that they were being charged an additional $16,000 over a period of 2 months for having accidentally activated CloudWatch, an AWS service that helps track metrics on performance.

This illustrates just how perilous the circumstances could be for companies who do not have a system in place that helps them prevent such mishaps.

Small Leaks Sink Big Ships

Unwanted expenses, if unnoticed, could quite possibly be catastrophic for companies using AWS who do not have a designated team that monitors spending on AWS on a regular basis.

This issue highlights the need for a feature like Anomaly Detection that allows companies with sizeable footprints on AWS to accurately identify and plug such 'leaks' in their budget.

Anomaly Detection – Key Features:

Anomaly Detection provides users with the information to:

- Identify excessive or irregular spending.
- Pinpoint changes in resource usage patterns and assess whether they were expected or unexpected.
- Be aware and in control of their expenses on AWS without having to check their account on a daily basis. Notifications are also sent to users via email to inform them of anomalies in their spending patterns.

Why Is Anomaly Detection Important?

Cost Optimization and Rapid Feedback

It goes without saying that the optimization of budgets is one of the key priorities for most AWS users. Anomaly Detection facilitates this optimization by alerting users and helping them discover anomalies in their spending as early as possible. Having a feature that automatically identifies anomalies early on allows users to not only save a lot of time in their monitoring efforts, but it also allows them to make the changes they need to prevent (or predict) such anomalies in the future.

Anomaly Recognition and Effective Cost Breakdowns

Sifting through and deciphering the invoices provided by AWS and then try to draw a correlation between costs and resource usage can be quite a demanding task. Not to mention that even if you were to recognize the fact that there's something irregular in your spending, you'd still need to be able to pinpoint the cause for this irregularity and figure out the changes you need to make to prevent unwanted spending in the future.

Fortunately, Anomaly Detection does most of the heavy lifting for users. Anomalies are automatically recognized and can be identified even by users with minimal technical expertise. Additional charts, graphs, and other visual tools provided by TrackIt's AI then allow the user to dive deeper into the cost analysis and make the necessary technical changes in AWS.

Accurate Monitoring of Costs for New Resources Launched on AWS

Anomaly Detection allows you to accurately monitor the evolution of costs when new resources are launched on AWS. An anomaly occurs as soon as the new resource is deployed, and this allows users to scrupulously track the changes in spending during the early stages after deployment.

An Algorithm Borrowed from The Financial Realm:

The algorithm for Anomaly Detection is based on Bollinger Bands, a type of statistical chart used in the financial industry to provide a relative definition of high and lows prices of a market. Bollinger Bands serve as an indicator designed to provide traders with information regarding price volatility. A Bollinger Band is typically a set of lines plotted two standard deviations (positively and negatively) away from a simple moving average of a specific security's price.

The algorithm's simplicity in terms of implementation and effectiveness in identifying abrupt changes in prices (or costs) allowed TrackIt's team to use it to effectually identify anomalies in spending on AWS.

How the Algorithm Works:

The time range of a month is split into smaller blocks of 24H (24 Hours) each. This gives us 30 blocks of 24H, with each block containing a series of values.

Then for each block (or interval) of 24H, all the values of the block are summed so that each block has one single value. This leads to exactly 30 values in total, one value per block/interval of 24H.

For each block, the moving average **μ** and the standard deviation **σ** are calculated on the last 3 blocks of 24H. (We take **n** = 3)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2}$$

This allows us to calculate an upper band and a lower band.

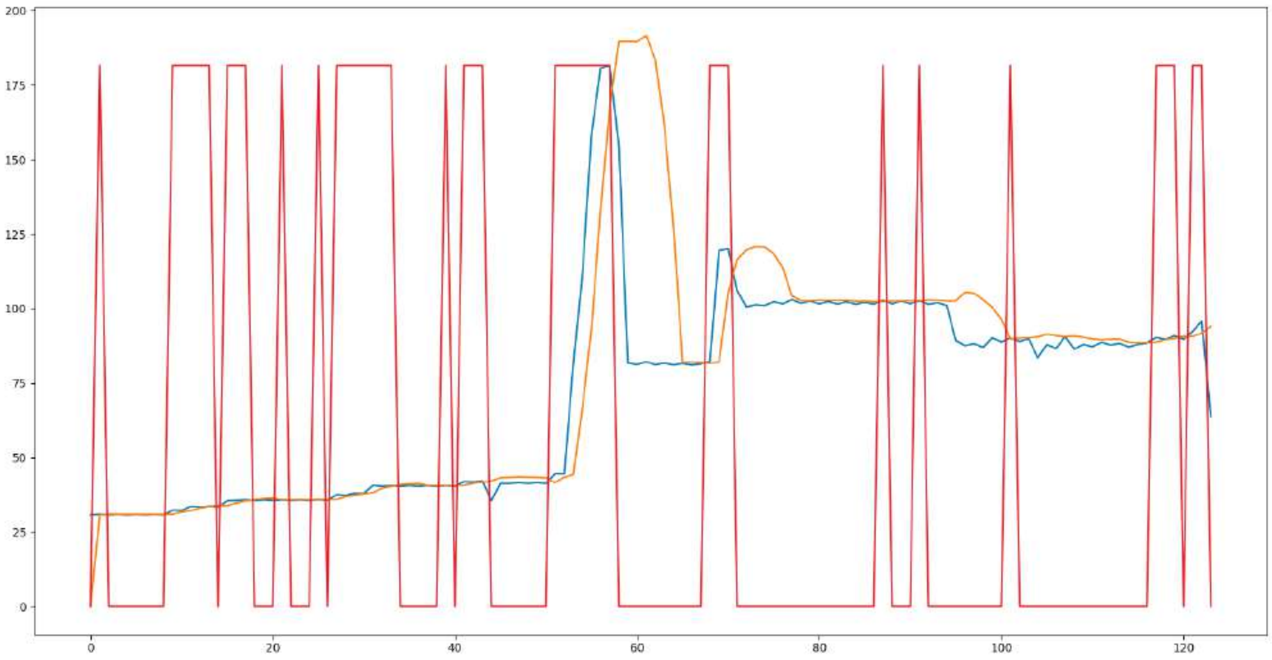
Upper band= μ+δσ

Lower band= μ-δσ

δ is the standard deviation coefficient. We took **δ** = 3.0. The value can be adjusted to have a better curve. If the value at a certain point in the graph exceeds the value of the upper band, we are alerted of a peak. Conversely, if the value at a certain point drops below the value of the lower band, we are alerted of a drop.

We will be focusing solely on the upper band which is of particular interest to us since it alerts us of peaks in cost.

Below is an example of a chart representing the AWS spending for the **Amazon EC2** service over 4 months. (So, we have 120 values in total.)



The real cost is represented in blue and the upper band is represented in orange. The red curve represents the alerts. Alerts come up when the blue curve exceeds the orange curve.

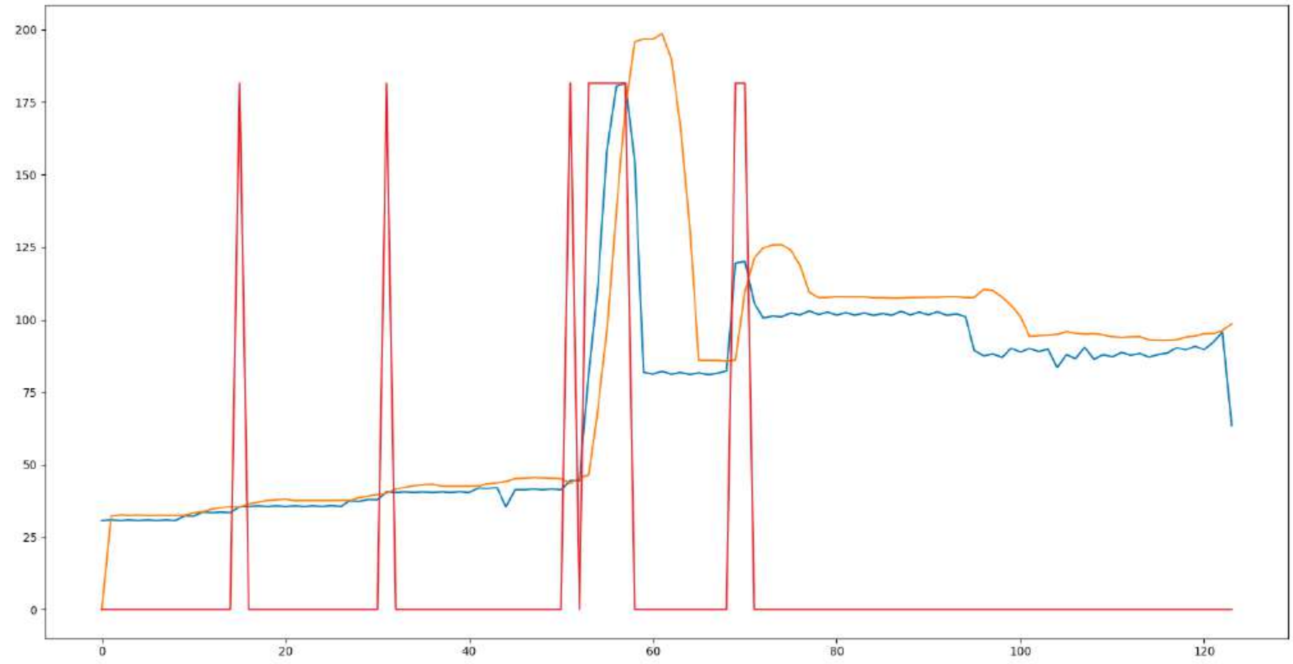
We add a margin of error of **5%** to reduce the sensitivity of the curve.

$$m=1.05$$

We then calculate the upper band again:

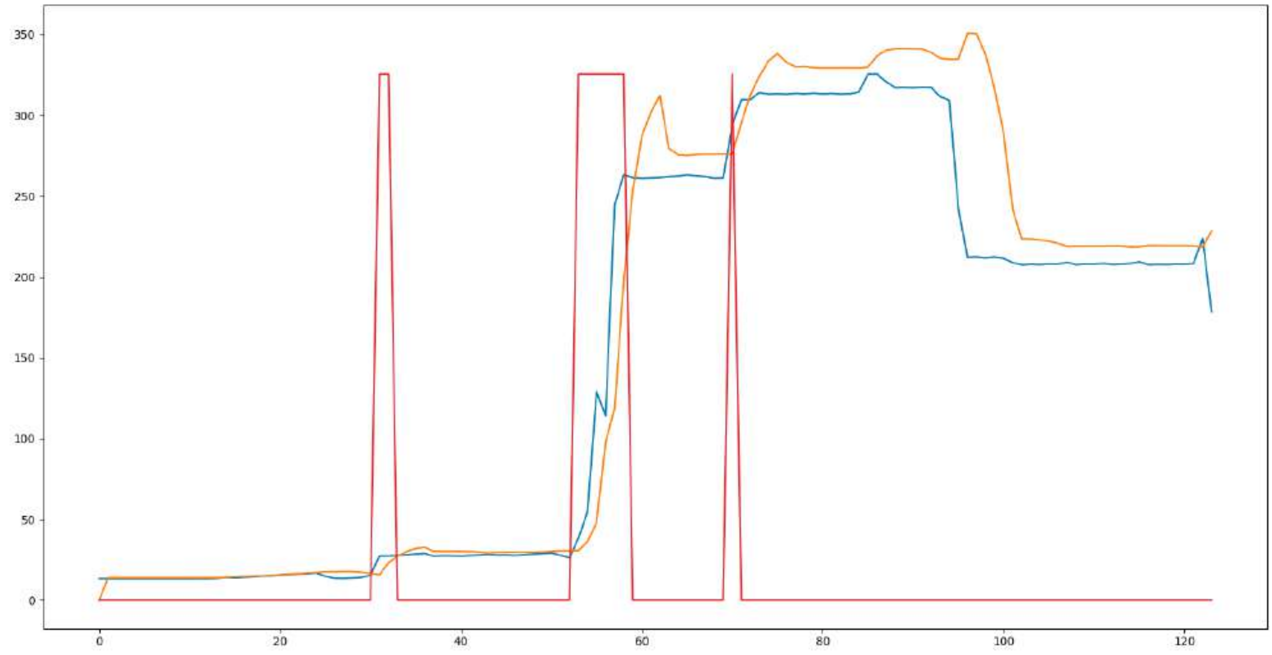
$$\mu m\text{-}\delta\sigma$$

With the new configuration, we can see that the alerts are more relevant. However, we still see useless efforts. Alerts were being triggered even for minor anomalies.



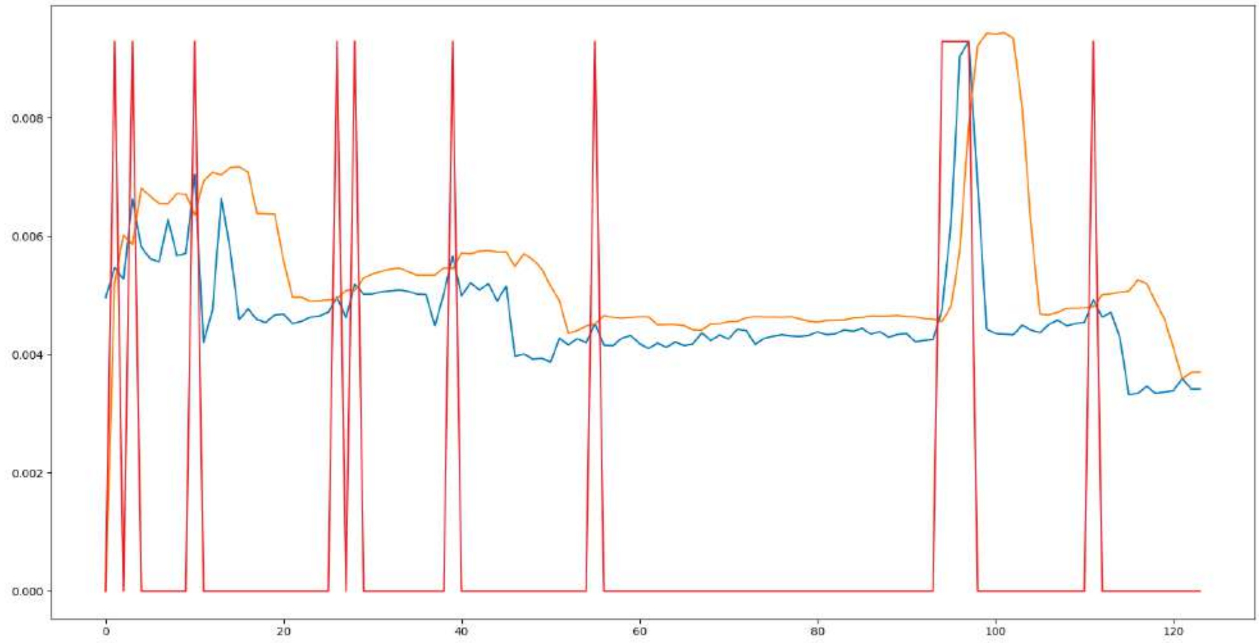
5 Anomalies are triggered here whenever the blue curve (real cost) exceeds the orange one (cost to not exceed).

Amazon RDS



Amazon RDS Chart - 3 anomaly alerts. Same algorithm: Bollinger Bands with the same margin of error.

Amazon Cloudtrail



Cloudtrail Chart: We notice anomalies that are not relevant at all - by taking a look at the y-axis of this chart, we notice that these anomalies are being triggered for minor increases in cost. (< $0.004)

Having noticed that AWS charges can be incredibly low in some cases, we ignore alerts related to costs that are below a certain amount (typically $20).

With all these manipulations, we keep only the most important and useful alerts.

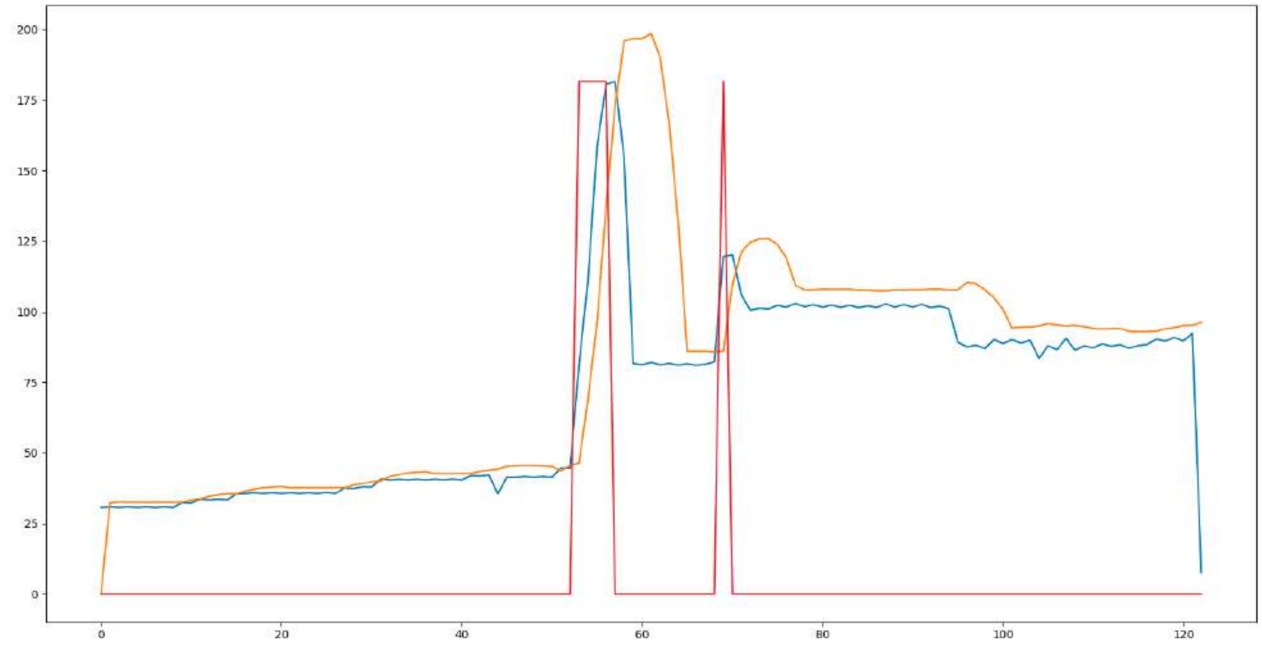
To make sure that the alerts being shown are the most relevant, we do the following:

- We take the 5 last days into consideration
- We sum the cloud costs for *each* service
- We keep the alert only if the service is in the top 5 services sorted by cost

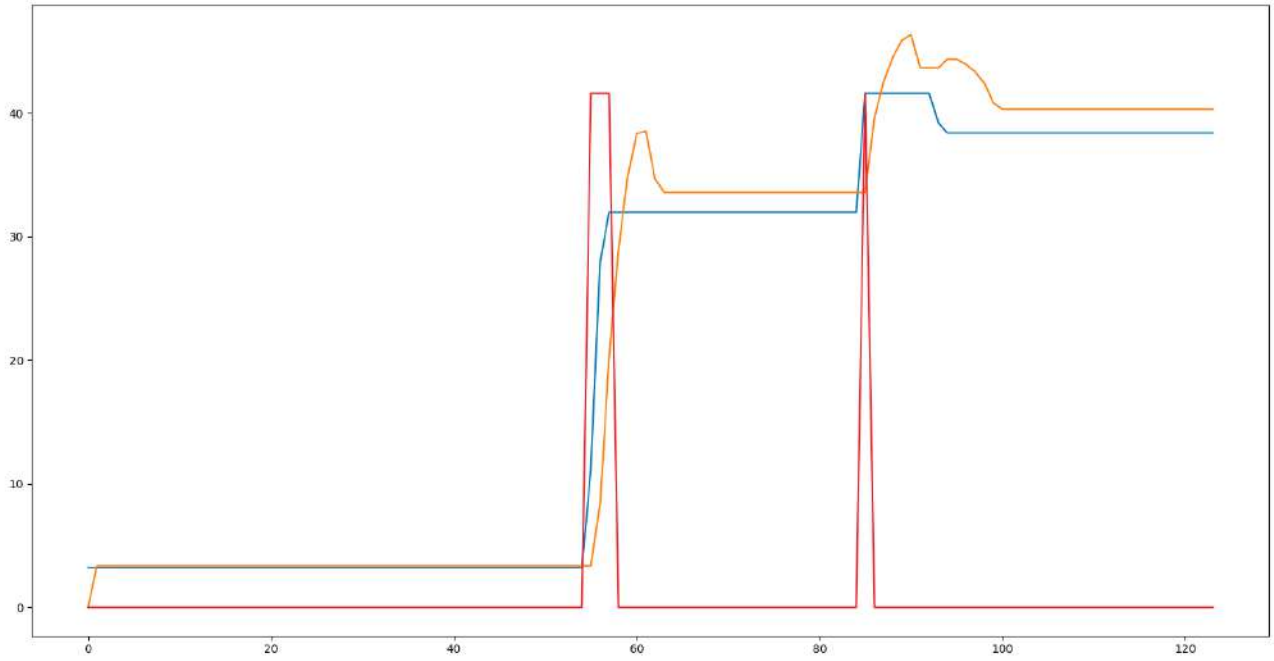
Furthermore, we also remove alerts if the cost is below 5% of the total bill of the day.

After taking these changes into consideration, we notice that only the most relevant anomalies - anomalies that are being triggered for increases in cost that are just big enough - are being shown:

Amazon EC2



Amazon RDS



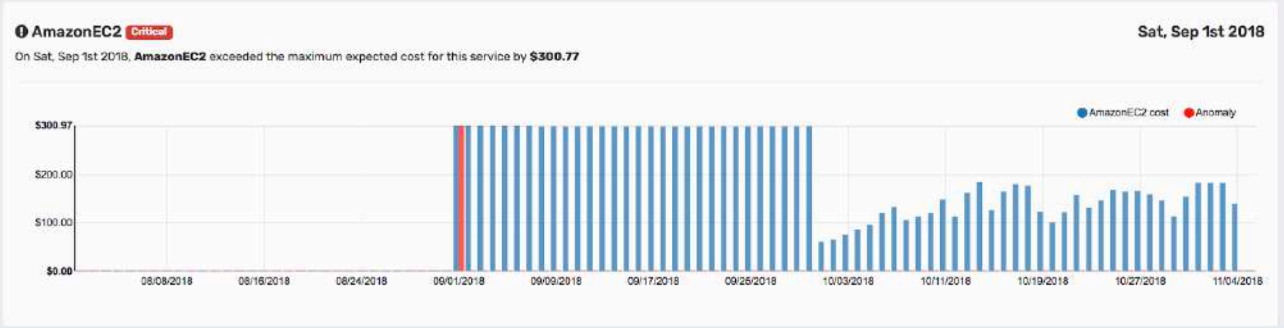
Key Benefits of Anomaly Detection:

Listed below are the key benefits of Anomaly Detection:

- **It saves users money** by helping them identify anomalies in their AWS expenditure.
- **It saves users time** by providing them with all the information they need to get to the root of an anomaly and figure out why it occurred.
- It helps users **identify unexpected costs** and provides them with the information needed to prevent these unnecessary expenses in the future.
- It allows users to accurately track and monitor the evolution of costs when new resources are deployed.

It enables users with a **minimal technical expertise** to identify anomalies and report it to their technical teams.

Example #1 - ElephantDrive



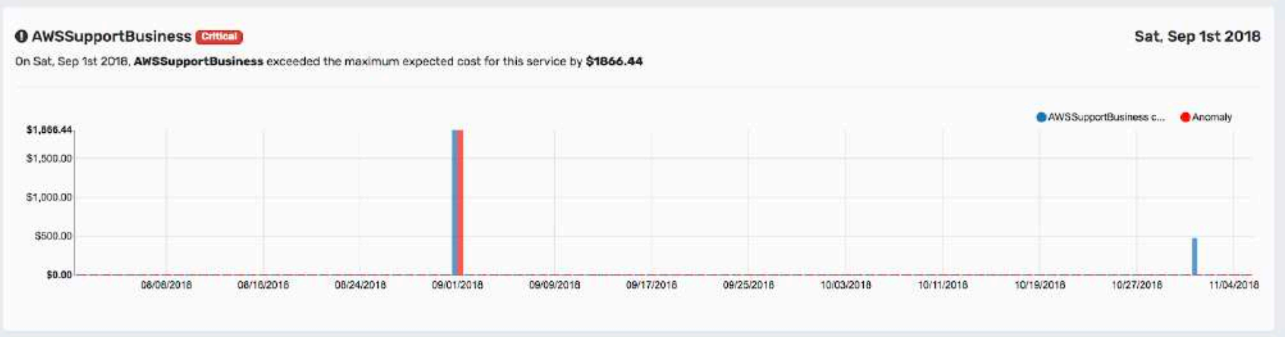
AmazonEC2 Critical

Sat, Sep 1st 2018

On Sat, Sep 1st 2018, **AmazonEC2** exceeded the maximum expected cost for this service by **$300.77**

Anomaly detected as soon as the cloud spend jumped from below $10 to $300.

Example #2 - Paradox



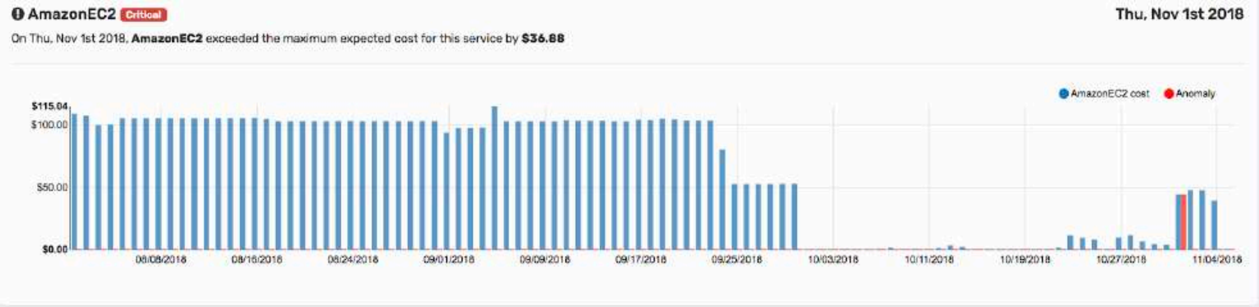
AWSSupportBusiness Critical

Sat, Sep 1st 2018

On Sat, Sep 1st 2018, **AWSSupportBusiness** exceeded the maximum expected cost for this service by **$1866.44**

Anomaly detected as soon as the cloud spend jumped from below $10 to $1868.

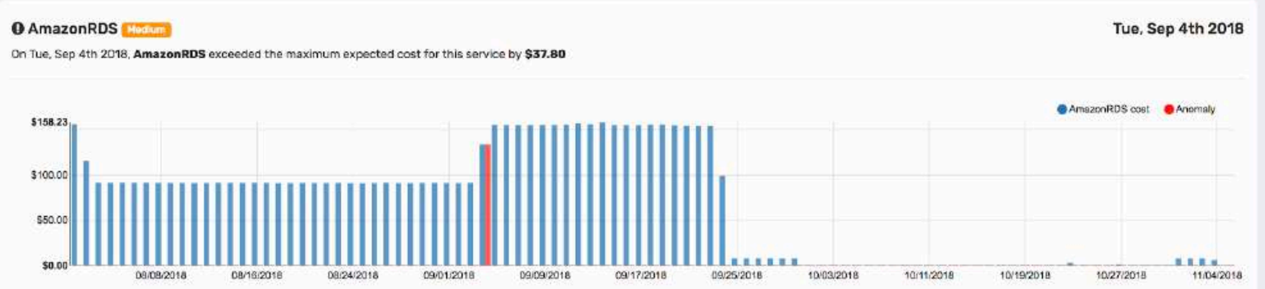
Example #3 - Delamaison



AmazonEC2 Critical

Thu, Nov 1st 2018

On Thu, Nov 1st 2018, **AmazonEC2** exceeded the maximum expected cost for this service by **$36.88**



AmazonRDS Medium

Tue, Sep 4th 2018

On Tue, Sep 4th 2018, **AmazonRDS** exceeded the maximum expected cost for this service by **$37.80**

Anomaly 1 detected as soon as the cloud spend jumped from $5 to $45.

Anomaly 2 detected as soon as the cloud spend jumped from $90 to $125.

References

https://www.researchgate.net/publication/233947082_Bollinger_Bands_Thirty_Years_Later
https://www.investopedia.com/articles/technical/102201.asp
https://www.iforex.in/bollinger-bands
https://www.bollingerbands.com