Amazon CloudWatchでSLOを監視してみた
CODT 2024 クロージングイベント版

2024-09-06

Cloud Operator Days Tokyo 2024

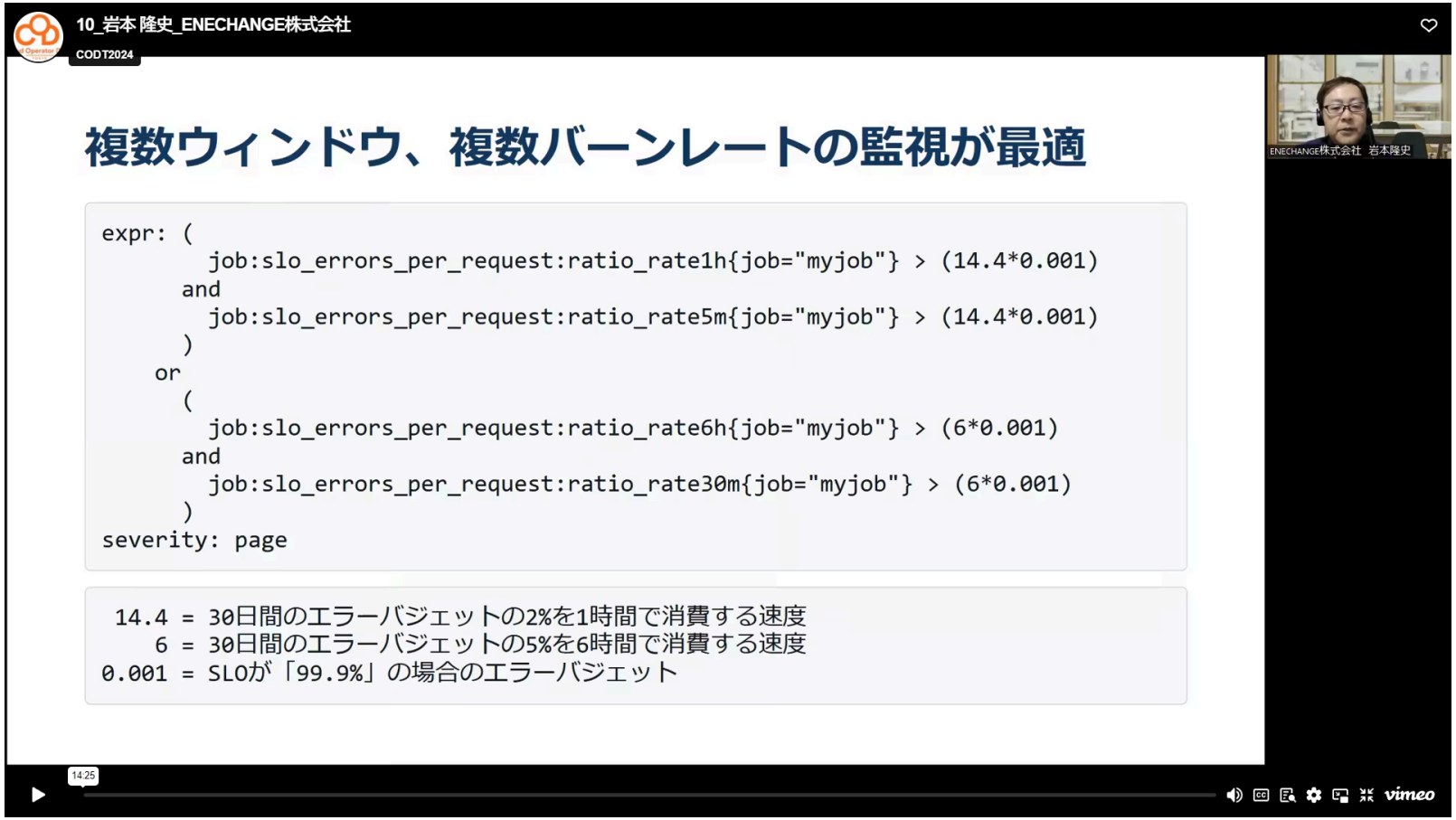
https://cloudopsdays.com/closing/

ENECHANGE株式会社 岩本隆史

岩本 隆史 / Takashi Iwamoto

- 現職：ENECHANGE株式会社 VPoT
- 前職：AWS Japan クラウドサポートアソシエイト
- AWS Community Builder (Cloud Operations)
- https://x.com/iwamot

セッション動画



https://event2024.cloudopsdays.com/2024/07/06/10/

Amazon CloudWatchでSLOを監視してみた

> 1. 複数ウィンドウ、複数バーンレートの監視が大前提
> 2. ENECHANGEではCloudWatchで監視中
> 3. 「複合アラーム」と「カスタムメトリクス」の活用が肝
> 4. コストは月20ドルほど

https://speakerdeck.com/iwamot/amazon-cloudwatch-slo-monitoring?slide=3

Q. なぜSLOの監視を始めたのか？

A.「オオカミ少年アラート」を減らすため

> 以前は、CPUやメモリの使用率といったリソースのメトリクスを直接的に監視しており、サービスが問題なく提供できていてもアラートが飛ぶ状況でした。いわゆる「オオカミ少年アラート」の多い状況です。

https://findy-tools.io/products/amazon-cloudwatch/36/197

指針：「原因」より「症状」にアラートを

> クラウドを使うにあたって、うまくアラートを設定するには SLI/SLO をきちんと定義し、それに応じたアラートの度合いを決めるのがよいです。また、SLO の対象となる指標としては、原因じゃなくて症状、つまりユーザへの具体的な影響を測れるものが良いです。そして、ある程度のエラーを許容しつつユーザへの影響がでないようなシステムの作りを目指しましょう。

https://medium.com/google-cloud-jp/alerting-in-cloud-deb0aa35ec16

成果：可用性やレイテンシの悪化に気づけるように

> オオカミ少年アラートが削減できた一方で、可用性やレイテンシの悪化にすぐ気づけるようになりました。また、それらの指標に対する開発チームの意識が高まりました。

https://findy-tools.io/products/amazon-cloudwatch/36/197

現時点の監視対象：7件のサービス、12件のURL

Terraformモジュール化により短時間で導入可能

```
module "slomon" {
  source = "git@github.com:enechange/terraform-modules.git//slomon-for-alb?ref=v0.53.0"

  environment_name              = "prod-enechange"
  alb_access_logs_s3_url        = local.alb_access_logs_s3_url
  sns_topic_names_for_paging    = ["cto-incident-enechange"]
  sns_topic_names_for_ticketing = ["cto-alert-enechange"]

  critical_user_journeys = {
    input1_submit = {
      http_method     = "POST"
      path            = "/try/input1_submit"
      dashboard_order = 1

      slo = {
        availability_target   = 95.0
        latency_p95_threshold = 4.0
        latency_p50_threshold = 3.0
      }
    }
  }
}
```

https://speakerdeck.com/iwamot/amazon-cloudwatch-slo-monitoring?slide=15

Q. CloudWatchを選んだ理由は？

A. 理想の条件式、かつ、低コストで監視できるから

ツール	理想の条件式	低コスト
CloudWatch	◯	◯
New Relic	△（検証当時）	△
Datadog	×（検証当時）	−

『ワークブック』の条件式が理想

```
expr: (
        job:slo_errors_per_request:ratio_rate1h{job="myjob"} > (14.4*0.001)
      and
        job:slo_errors_per_request:ratio_rate5m{job="myjob"} > (14.4*0.001)
      )
    or
      (
        job:slo_errors_per_request:ratio_rate6h{job="myjob"} > (6*0.001)
      and
        job:slo_errors_per_request:ratio_rate30m{job="myjob"} > (6*0.001)
      )
```

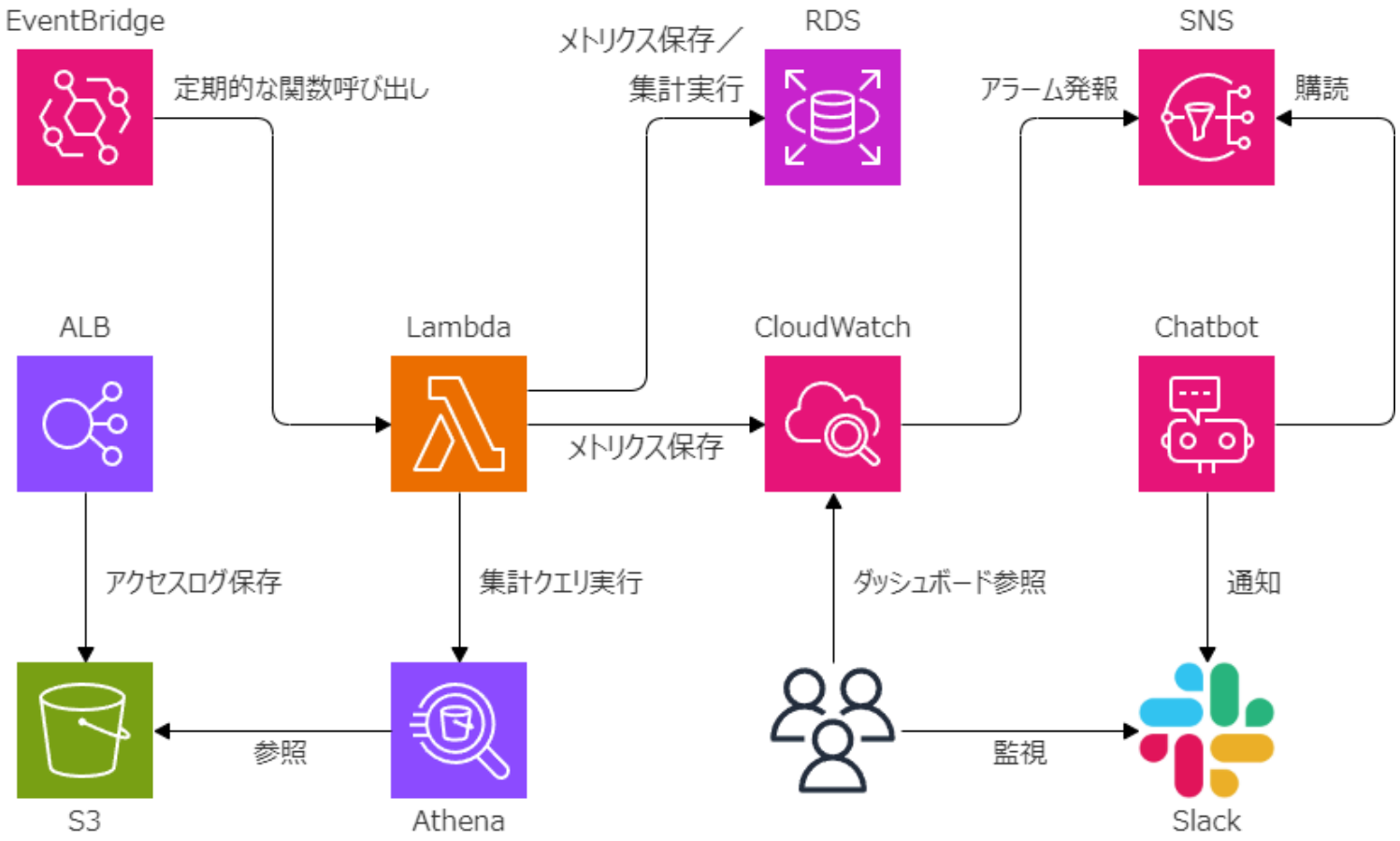
https://sre.google/workbook/alerting-on-slos/

CloudWatchなら、1件20ドル/月で監視可能

課金対象	件数	コスト (ドル/月)
ダッシュボード	1	3.0
メトリクス	28	8.4
アラーム	17	1.7
複合アラーム	5	2.5

https://speakerdeck.com/iwamot/amazon-cloudwatch-slo-monitoring?slide=24

実装もそれほど難しくなかった



https://findy-tools.io/products/amazon-cloudwatch/36/197

Amazon曰く「倹約は創意工夫、自立心、発明の源」

> **Frugality**
>
> 私たちは少ないリソースでより多くのことを実現します。倹約の精神は創意工夫、自立心、発明を育む源になります。スタッフの人数、予算、固定費は多ければよいというものではありません。

https://www.amazon.jobs/content/jp/our-workplace/leadership-principles

Q. 今後の展望は？

A. ツールの発展に応じて柔軟に

> ローリングウィンドウ機能がCloudWatchに実装されれば作り込みが減らせるので、AWSに機能追加の要望を出そうと考えています。
>
> 一方で、New RelicやDatadogでも柔軟な実装が可能になれば、それらのツールに移行することもありえます。CloudWatchに比べ、サービスレベル低下の原因調査がしやすくなるためです。

https://findy-tools.io/products/amazon-cloudwatch/36/197

ご清聴ありがとうございました

- Amazon CloudWatchでSLOを監視してみた
- なぜSLOの監視を始めたのか？
  - 「オオカミ少年アラート」を減らすため
- CloudWatchを選んだ理由は？
  - 理想の条件式、かつ、低コストで監視できるから
- 今後の展望は？
  - ツールの発展に応じて柔軟に