

JAWSUG 名古屋

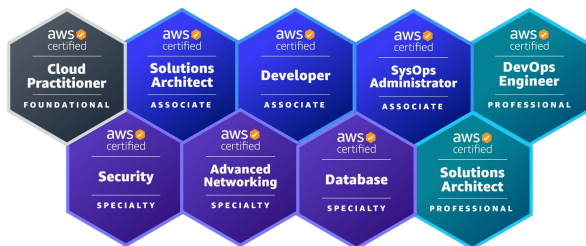
CloudWatchカスタムメトリクスを
トリガにスケールするEC2 ASG

2023/7/11

氏名:新川 貴章

AWS/Azure インフラエンジニア

- iretテクニカルアンバサダー 2023
- ブログ(oji-cloud)
- 2021年4月 WEB+DB PRESS記事執筆



オブザーバビリティとは

オブザーバビリティとは

- オブザーバビリティ(Observability) → 可観測性と訳される。
- モニタリングは、「何が起きているか」を見ること。
- オブザーバビリティは、「何が障害の原因だったのか」、「どこに影響があるのか」「どう対処すれば良いのか」を把握すること。



今日伝えたいこと

CloudWatchメトリクスだけでは
「なぬ。なんかおこった！」



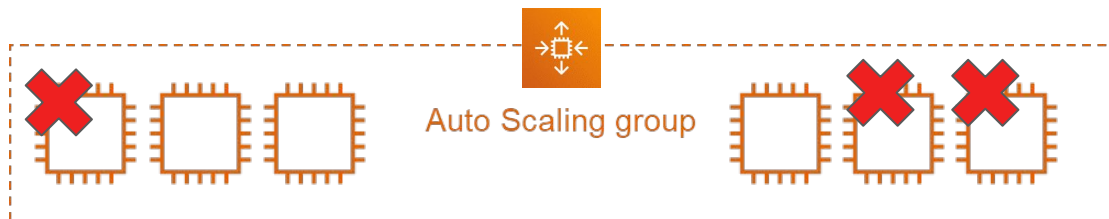
しか見えなかった

CloudWatchをもっと観測できるように
カスタマイズして
さらに自動で対処できるようにしたぜ！



私が遭遇した事象

- とあるEC2 Auto Scaling構成の httpサーバー(Apache)
- 定期的に、一部インスタンスでElastic Load Balancing のヘルスチェックのunhealthy が起こるようになった
- 特定のインスタンスではなく、事象発生時に複数台で起き始める
- メトリクス,ログから原因不明→時間の経過で解消される



原因と対処

- unhealthy のインスタンス停止前にEC2に乗り込んで調査
- ApacheのWorkersにはIdleは見受けられず、スレッドは使い切っていることが判明(httpサーバーが処理しきれない状態に)
- 性能改善が必要と判断
 - Apacheの同時接続数(MaxRequestWorkers)の変更
 - ApacheのKeepAliveTimeout値の見直し
 - インスタンスの台数増やす、単純に増やすとコスト増
 - EC2 ASG使いたいけど、台数増やすタイミングが不明

手作業の監視と対処を何とかしたい！

- **Apacheプロセス数を可視化したい！**
 - EC2に乗り込んで、`ps -ef | grep http` を打ちたくない。
- **Webサーバーの限界を事前に対処したい！**
 - プロセス数がMaxに近づいたらASGをスケールアウト？ 安定したらスケールイン？

だけど、CloudWatch にプロセス数を見るメトリクスはない？ プロセス数を可視化したい！

ASG単位のメトリクスで、(自動で)スケールアウトさせたい！

CloudWatch Agent カスタムメトリクスを活用①

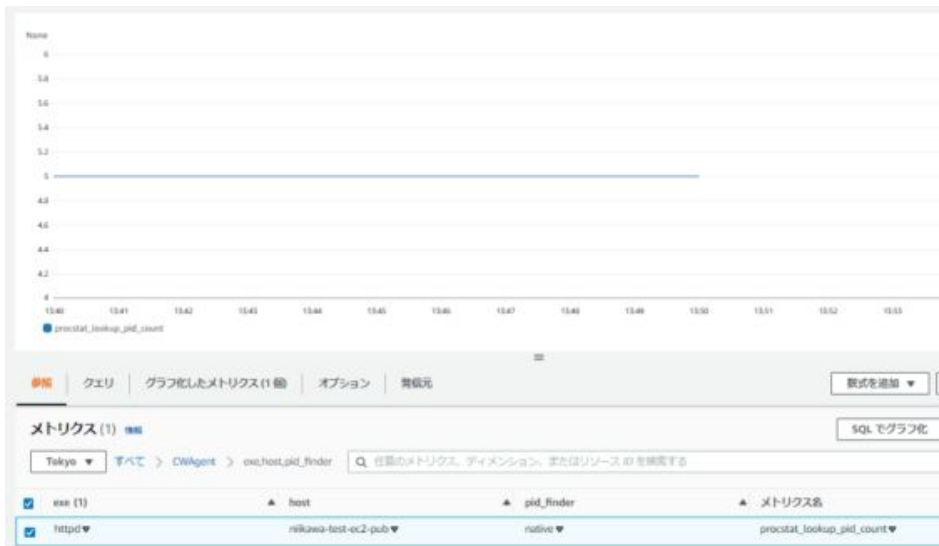
- amazon-cloudwatch-agent は、procstat を利用して、cpu_time, cpu_usage, memory, pid_count などのメトリクスが収集できる
- プロセスのダウンを検知するケース(例: sshdのプロセスがダウンした)、プロセス数上限を監視するケース(例: アクセス増によりhttpdのプロセスが100個起動している)などに対応可
- アラームを設定して、プロセス数がしきい値を上回った or 下回った場合に、通知を行う

CloudWatch Agent カスタムメトリクスを活用①

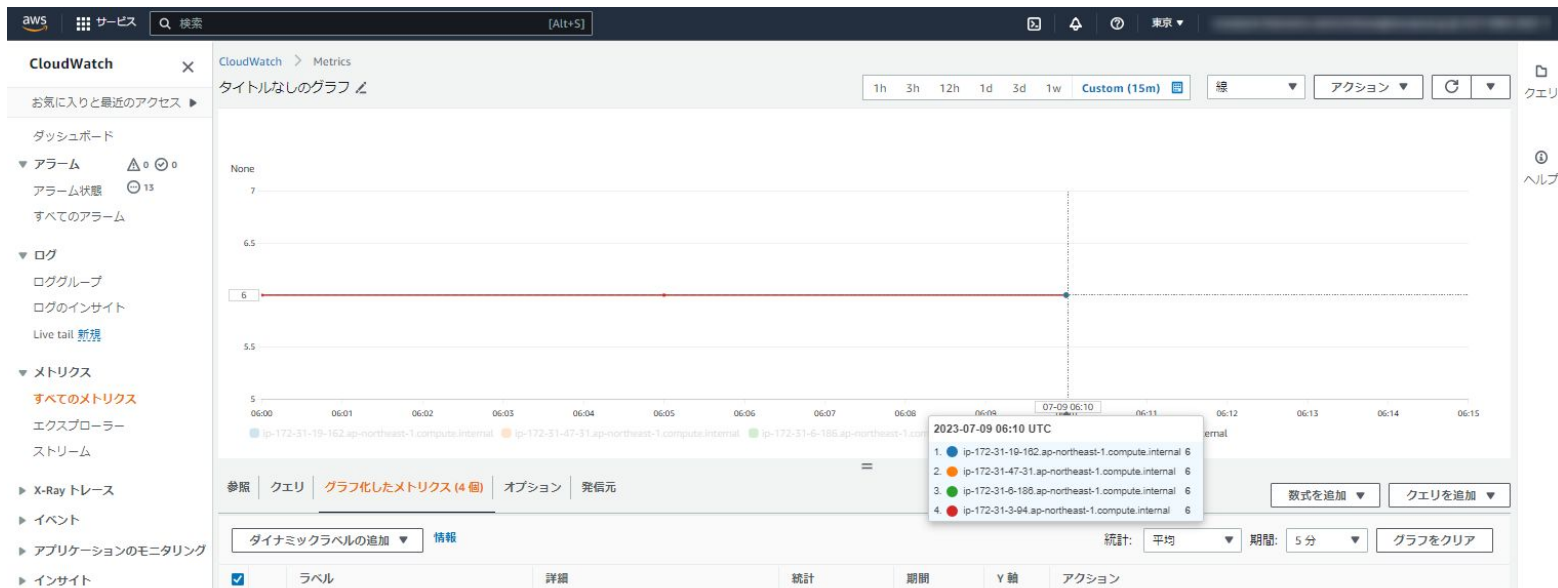
```
$ cd /opt/aws/amazon-cloudwatch-agent/etc
$ sudo vi amazon-cloudwatch-agent.json

$ sudo systemctl restart amazon-cloudwatch-agent
$ sudo systemctl status amazon-cloudwatch-agent
```

```
{
  "metrics": {
    "metrics_collected": {
      "procstat": [
        {
          "exe": "httpd",
          "measurement": [
            "cpu_time",
            "pid_count"
          ]
        }
      ]
    }
  }
}
```



CloudWatch Agent カスタムメトリクスを活用①



待てよ？ 可視化できたのは良いけど、
AutoScalingポリシーのトリガに使えない？
インスタンスの台数分スケールアウトが発動しちゃうかも！？

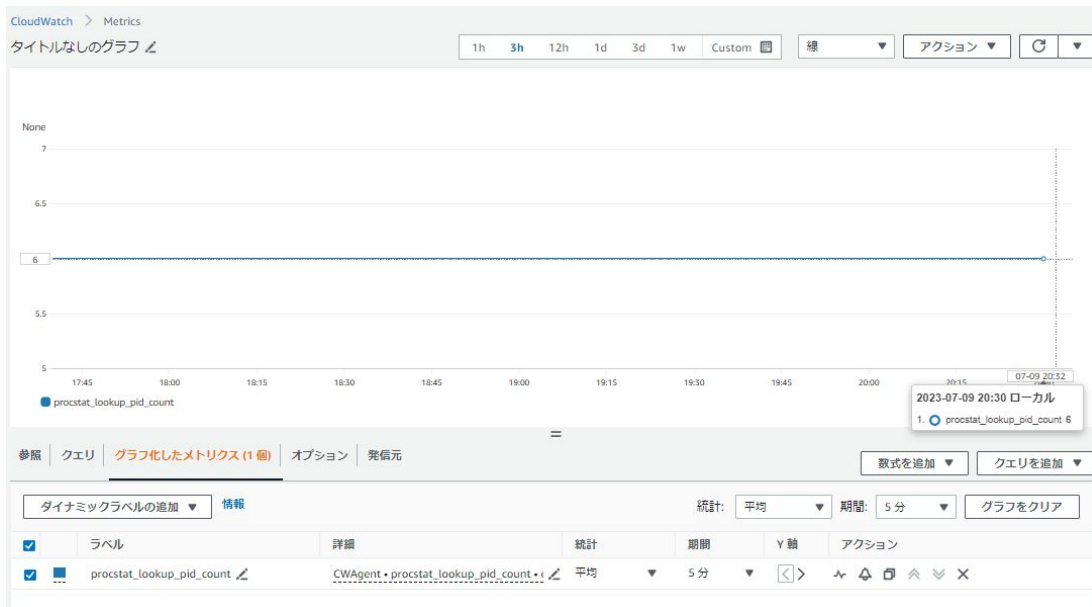
CloudWatch Agent カスタムメトリクスを活用②

```
{
  "metrics": {
    "append_dimensions": {
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}"
    },
    "aggregation_dimensions": [{"AutoScalingGroupName"}],
    "metrics_collected": {
      "procstat": [
        {
          "exe": "httpd",
          "measurement": [
            "pid_count"
          ],
          "metrics_collection_interval": 60
        }
      ]
    }
  }
}
```

インスタンス単位のメトリクスではなく、**aggregation_dimensions**でメトリクスを集約できるぞ！

ここがポイント！

- EC2 の権限を要チェック
- `metrics_collection_interval` は付けるべし
- `amazon-cloudwatch-agent` の `systemd` 設定カスタマイズのスレ (ASG利用時の場合)



CloudWatch Agent カスタムメトリクスを活用②

参照 | クエリ | **グラフ化したメトリクス (1個)** | オプション | 発信元

ダイナミックラベルの追加 | 情報

統計: 平均 | 期間: 5分 | グラフをクリア

ラベル	詳細	統計	期間	Y軸	アクション
procstat_lookup_pid_count	CWAgent + procstat_lookup_pid_count + r	平均	5分		アラームの作成

メトリクス

編集

グラフ
このアラームは、5分内の1データポイントで上回る赤線を越える場合に、トリガーされます。

名前空間
CWAgent

メトリクス名
procstat_lookup_pid_count

AutoScalingGroupName
niikawa-test-webserver-ASG

統計
平均値

期間
5分

動的スケーリングポリシーを作成する

ポリシータイプ
シンプルスケーリング

スケーリングポリシー名
ASPolicy_httpd_proc_high_level

CloudWatch アラーム
次の場合にいつでも容量をスケールできるアラームを選択します。
httpd_proc_high_level_alarm

CloudWatch アラームを作成する
アラームのしきい値を超過: 次のメトリクスディメンションに対して 300 秒間の 1 連続期間で procstat_lookup_pid_count >= 10:

AutoScalingGroupName = niikawa-test-webserver-ASG

アクションを実行する
追加 | 1 | 容量ユニット

次に、待機します
600 | 別の規模の拡大や縮小を許可するまでの秒数

キャンセル | 作成

ASGポリシーの最大数指定に注意、スケーリングの条件指定に注意

条件

しきい値の種類

静的
値をしきい値として使用

異常検出
バンドをしきい値として使用

procstat_lookup_pid_count が次の時...
アラーム条件を定義します。

より大きい
> しきい値

以上
>= しきい値

以下
<= しきい値

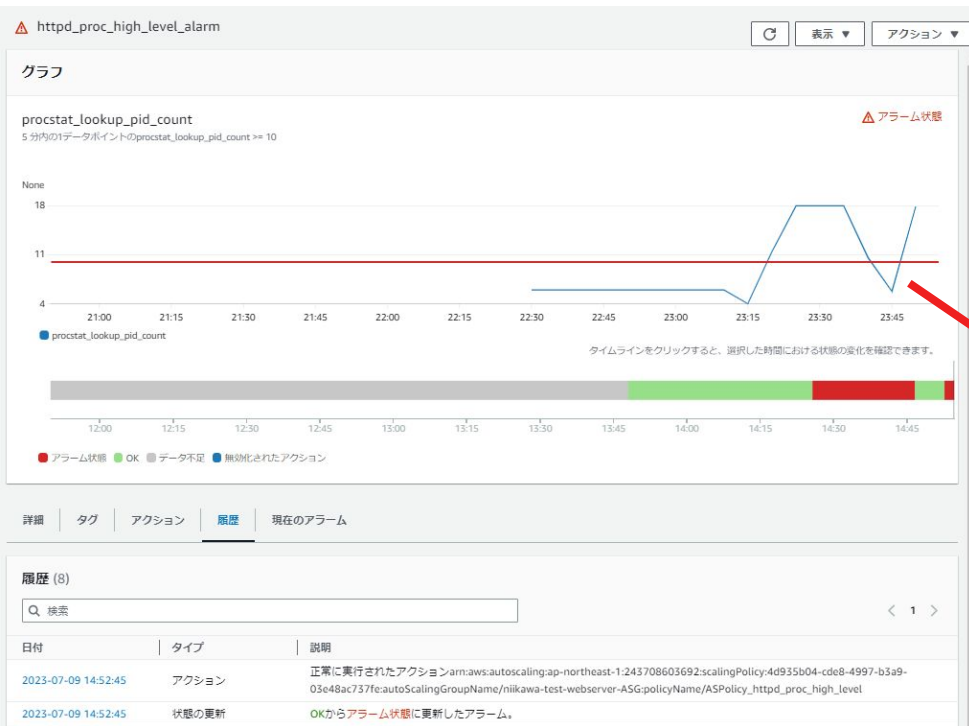
より低い
< しきい値

... よりも
しきい値を定義します。

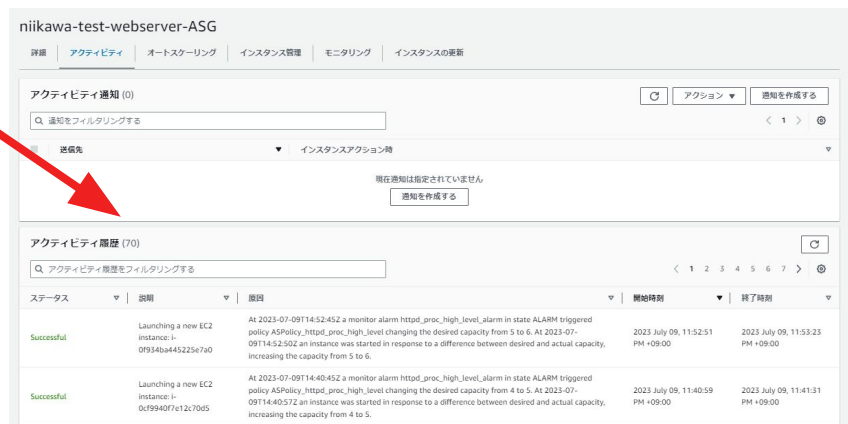
10

数字である必要があります

CloudWatch Agent カスタムメトリクスを活用②



プロセス数がしきい値を超えると、期待通り、スケールアウト発動！



今日伝えたいこと (おさらい)

CloudWatchメトリクスだけでは
「なぬ。なんかおこった！」



しか見えなかった

CloudWatchをもっと観測できるように
カスタマイズして
さらに自動で対処できるようにしたぜ！

