

StackStormハンズオン 障害自動検知と自動修復(AR)

幕間: Auto Remediationについて

NTTテクノクロス株式会社
クラウド&セキュリティ事業部
萬治 渉 <manji.wataru@po.ntt-tx.co.jp>

はじめに

このセッションではvSRXへのARハンズオンの事前説明として

- Auto-Remediationが注目された背景と事例
 - StackStormによるAR
- について、約20分で話します。

目次

1. Auto-Remediationについて

- 注目された背景
- 各社の事例

2. StackStormによるARの実現

- Auto-Remediationの構成要素
- 従来の自動化と問題点
- StackStormによる解決
- 動作イメージとハンズオン内容について

Auto-Remediationについて



自動復旧の流行の背景と各社の導入事例

Auto-Remediation ?

機器やソフトウェアのエラーを
自動的に解決、復旧するという概念。

運用自動化という概念の一部であるが、
その中でも大きな比率を占める。

注目された背景

2000年代以降、彼らは実感した。

「人手で運用できないほど、サービスが大きくなった」
...そんな企業が続々と現れる。

Google, Facebook, Twitter, Apple, etc.

彼らの悩みはおおよそ以下。

- 運用に人を取られ、開発と改善に予算を回せないこと
- 毎晩鳴り響く障害アラートで担当者が疲弊すること

注目された背景

そんな中、2011年にFacebookによる
「**Making Facebook Self-Healing**」

<https://www.facebook.com/notes/facebook-engineering/making-facebook-self-healing/10150275248698920/>

という記事が発表され、

「障害対応の自動化 (Auto-Remediation)」
という言葉が世間に出た。



注目された背景

「運用業務のソフトウェア化&自動化」

によって、サービスの運用が以下のように変わる。

- サービス規模と運用稼動が比例しない
⇒ 開発、改善によりリソースを割ける
- 必要なときにしか夜に人を起こさない
⇒ 担当者の体力と人権を確保できる

各社の事例

Google:

「SRE本」でも話題になったように、
運用の自動化を軸にして以下の方法論を確立。

- 開発速度と信頼性のバランスを
取るための方法論
- より運用自動化を簡易にするための
アーキテクチャ

上記の方法論に従って、

信頼性の高いサービスを早い周期で提供している

各社の事例

Facebook:

サーバ機器などのセットアップ、運用を
極力自動化し、2015年には

「サーバ25,000台あたり1人の技術者しか必要ない」
というレベルの省力化を成し遂げている。

An Inside Look at Facebook's Approach to Automation and Human Work より

<https://hbr.org/2015/06/an-inside-look-at-facebooks-approach-to-automation-and-human-work>

次に、

- **Auto-Remediationを実現するための要素**
- **StackStormによる基本的な実装方法**

について説明していきます

StackStormによるARの実現



Event-Driven & Workflow + 拡張性

まず伝えたいことを3行で

1. StackStormは
2. ARなどの運用自動化を実現するために
3. 必要十分な機能を備えています！



Auto-Remediationの構成要素

故障対応には大きく次のような段階がある

1. 故障の検知
2. 事象の把握
3. 原因の特定
4. 修正や解決
5. 復旧の確認

故障対応の自動化を実現するには、
上記のステップを実施するシステムが必要

一応ソフトウェア化できるはず

1. 故障の検知 ⇒ 監視ツール
2. 事象の把握 ⇒ 監視ルール | スクリプト
3. 原因の特定 ⇒ 監視ルール | スクリプト
4. 修正や解決 ⇒ スクリプト
5. 復旧の確認 ⇒ スクリプト(通知含む)



従来の自動化と問題点

無料でやるとしても色々手段はある。

例) Zabbixのカスタムスクリプトを使う

しかし、

- スクリプトの部品化が難しく、
複雑になりやすい

などの問題がある。

StackStormの機能で解決

StackStormは

「操作手順の部品化、記述の簡易化」
を実現する機能を備えている

- 「Workflow」と「Action」の役割分担
- YAMLベースで書きやすい「Workflow」

また、ActionやWorkflowを
「パッケージにして共有する」
機能を備えている。

詳しく書くと

- 操作スクリプトを分割して、
Workflowという形にまとめることができる
⇒ 部品化とその実行を管理できる！
長い手順も簡易に記述できる！
- 部品化された操作スクリプトを簡単に
共有&インポートできる
⇒ スクリプトの自作をしなくてよい！

という点で、StackStormを使うメリットがある

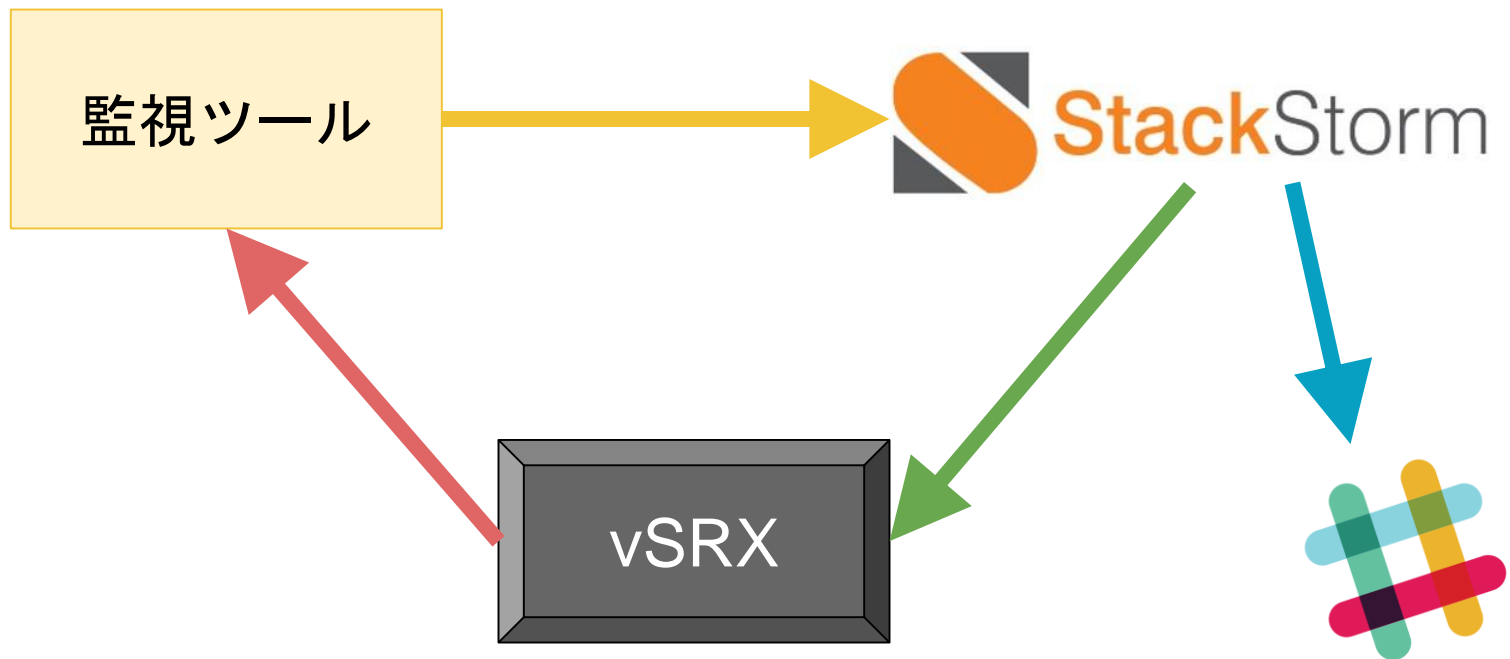
ARの構成要素との対応。必要十分ですね。

1. 故障の検知 ⇒ 監視ツール
2. 事象の把握 ⇒ 監視ルール | st2 rule
3. 原因の特定 ⇒ 監視ルール | st2 rule
4. 修正や解決 ⇒ st2 Workflow & Action
5. 復旧の確認 ⇒ st2 Workflow & Action



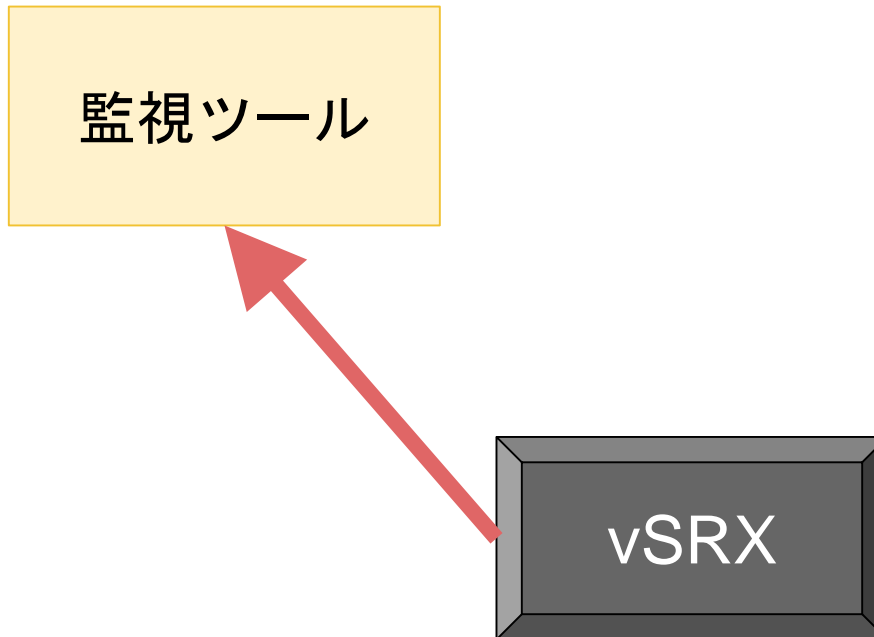
AR動作イメージとハンズオンでの解説ポイント

ここからは図で解説。
以下のようなイメージ。



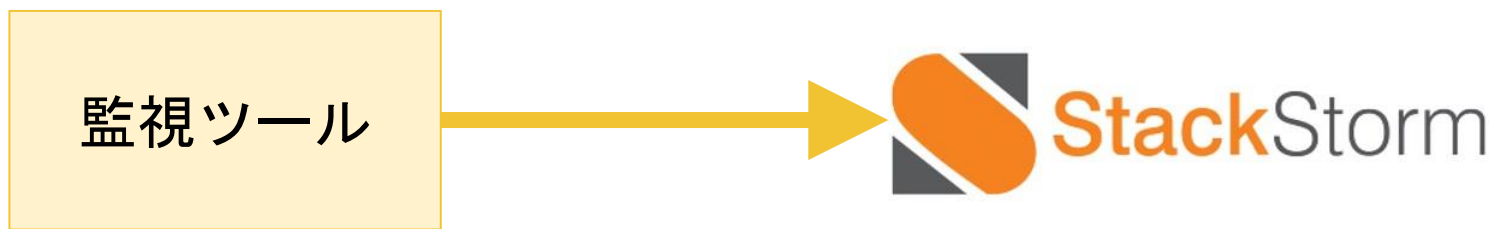
その1

監視ツールはvSRXから
定期的に稼動情報を収集する



その2

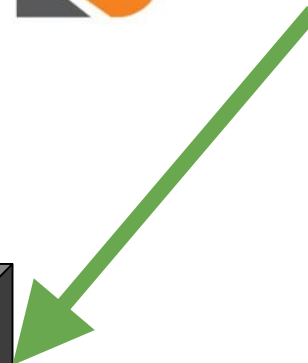
監視ツールはst2に稼動情報を転送する。
本ハンズオンでは「障害発生時のみ」ではなく、
「正常動作している状態」でも情報を転送するものとする。



その3

st2は監視ツールから受け取った情報を元に、
ruleに従って所定のWorkflow | Actionを実行
する

本ハンズオンでは
1. 単一のActionを実行
2. Workflowの実行
の2パターンを順に実習します



その4

st2からSlackなどへ通知を行なう

本ハンズオンでは、
Workflowによって
「復旧操作時にSlack通知を行なう」
方法について実習します

