

誰がOpenStackを見張るのか？

# OpenStackを監視・管理する コンポーネントたち

---

NTTソフトウェア株式会社

クラウド&セキュリティ事業部 第一事業ユニット

本上 力丸

## 発表者について

- 本上カ丸
  - NTTソフトウェア株式会社所属
  - Githubアカウント:rkmrHonjo



- EssexリリースからOpenStackクラウドの設計、検証、コントリビュート活動に携わる。
- 特にディープに関わってきたコンポーネントはNova, Cinder。
- 新しいコンポーネントの隆盛にも興味をもち、ウォッチ中。

## セッションの流れ

---

- セッションの目的
- OpenStackを監視するコンポーネントの紹介
- OpenStackを管理するコンポーネントの紹介
- 注目Blueprintsの紹介
- まとめ、およびこれからの展望

# OpenStack監視・管理 今昔

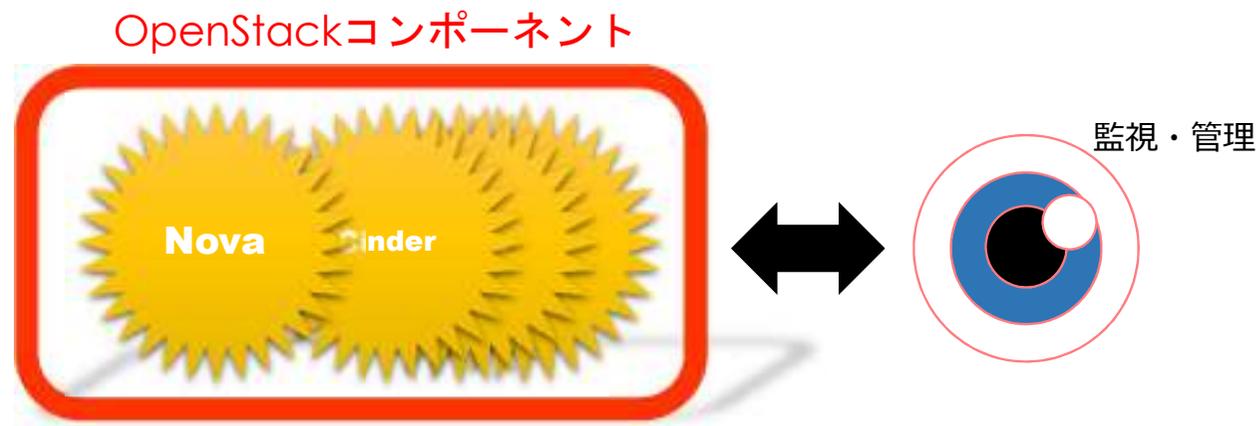
- OpenStackの監視、管理を行うソフトウェア

- **昔：**

コンポーネントとしてはceilometer、heat程度で、高度な目的にはOpenStack外のアプリケーションを利用。

- **今：**

監視・管理ソフトウェア自体もOpenStackコンポーネントとして実現。



# OpenStack監視・管理 今昔

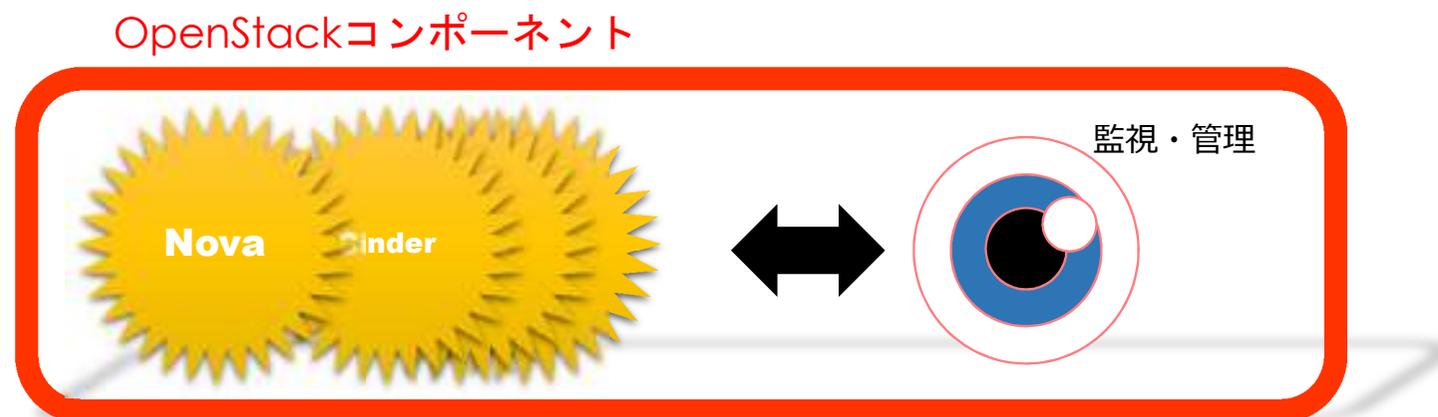
- OpenStackの監視、管理を行うソフトウェア

- 昔 :

コンポーネントとしてはceilometer、heat程度で、高度な目的にはOpenStack外のアプリケーションを利用。

- 今 :

監視・管理ソフトウェア自体もOpenStackコンポーネントとして実現。



## セッションの目的

---

- 監視
  - OpenStackの各種情報の収集や計測をしたり、アラームを挙げたりする
- 管理
  - OpenStackコンポーネントに命令を出して各種操作をする

上記のはたらきをするプロジェクトが次々Big Tent入りしている。  
このセッションでは、監視・管理を行うプロジェクトそれぞれの現状や差異、今後について紹介します。

## Big Tent入りしている監視コンポーネント一覧

プロジェクト名		提供サービス
Congress		ポリシー適用サービス
Monasca		モニタリングサービス
Telemetry	Ceilometer	計測サービス
	Aodh	
	Gnocchi	
Vitrage		根本原因解析サービス

## 新参監視系コンポーネント – 簡単な紹介

- **Congress**

- OpenStack仮想資源やVMのゲストOS上のアプリケーションに対して、定義したポリシー（例：ビジネス要件から導かれる仮想資源同士の関係性、セキュリティ要件、など）に準拠しているか、チェックができる。

- **Monasca**

- OpenStackのプロセス自体や、プロセスが稼働するホスト、またアプリケーション（例：MySQLなど）等のモニタリングができる。

- **Vitrage**

- Root Cause Analysis（根本原因解析）を行う。
- 具体的には、OpenStack資源同士やそのホスト等の関係性を明確化したり、問題が発生した場合にその根本原因がどこにあるかの推測をする。

## 監視系コンポーネント間の比較 - 一覧

		Congress	Monasca	Telemetry	Vitrage
正式リリース (BigTent入り) 時期		Kilo	Mitaka	Havana (Ceilometer)	Newton
ターゲット	OpenStackプロセス含むインフラ	○	○	-	○
	OpenStack資源	○	○	○	○
	ゲストOS上の資源、アプリ	○	○	△	○
主な対象利用者		管理者 /ユーザ	管理者 /ユーザ	管理者	管理者

7/6現在の状況に基づく

# 従来監視系コンポーネント（Ceilometer）とそれ以外の差異

- 監視 – Telemetry(Ceilometer)と他は何が異なるのか？
  - **Congress**
    - Congressで規定するポリシーは表現力が高く、Ceilometerのアラーム条件以上にきめ細やかで、かつ広い範囲に適用ができる。
    - 前述のMonasca, Vitrageと違い、Congressはチェック結果に応じて各コンポーネントにAPI発行するという事もできる。  
(メインはポリシーによるチェックなので便宜上監視側に入れた)
  - **Monasca**
    - OpenStack自体のプロセスや、当該プロセスのホスト、関連アプリケーションの等Ceilometerの対象外の監視ができる。
    - 利用者をシステム管理者に限定していない。
  - **Vitrage**
    - 1つの問題によって連鎖的に複数のアラームが発生するような場合に、根本的に何が原因となっていそうか？の交通整理をしてくれる。

## Big Tent入りしている**管理コンポーネント**一覧

プロジェクト名	提供サービス
Heat	オーケストレーションサービス
Mistral	ワークフローサービス
Senlin	クラスタリングサービス
Watcher	インフラ最適化サービス

## 新参管理コンポーネント – 簡単な紹介

### • Mistral

- OpenStackに対する操作の流れをワークフローとして記述し、cron形式の自動実行や通知契機の実行ができる。
- Amazon SWF相当。

### • Senlin

- OpenStack仮想資源（novaのVMなど）のクラスタリング化と、クラスタに対するロードバランシング、高可用等のポリシー（Congressのポリシーとは別）の適用ができる。

### • Watcher

- 仮想資源の物理資源利用状況（例：VMのCPU利用率）等を監視し、系としての最適化を行い、総所有コスト（TCO）を下げる。

## 管理系コンポーネント間の比較 - 一覧

		Heat	Mistral	Senlin	Watcher
正式リリース (BigTent入り) 時期		Havana	Kilo	Mitaka	Newton
ターゲット	OpenStackプロセス含むインフラ	-	-	-	○
	OpenStack資源	○	○	○	○
	ゲストOS上の資源、アプリ	○	○	-	-
主な対象利用者		管理者 /ユーザ	管理者 /ユーザ	管理者 /ユーザ	管理者

7/6現在の状況に基づく

# 従来管理系コンポーネント（Heat）とそれ以外の差異(1)

- 管理 – Heatと他は何が異なるのか？
  - **Mistral**
    - Mistralで利用するワークフローは、条件分岐など、**振る舞いを記載できる**。
    - また、例えばVMを一日おきに再起動するなどの定期処理もMistralで実現できるタスクである。
    - Mistralのワークフローの中にheatのstack実行を組み込むこともできるし、逆にHeatのテンプレートにMistralのワークフローをリソースとして指定することもできる。
  - **Senlin**
    - 元々はHeatのオートスケーリング機能強化のBlueprintとして始まった。
    - **本質は資源のクラスタリング**で、オートスケーリングはクラスタリングによって実現できる機能の1つに過ぎないという結論になり、独立した。

## 従来管理系コンポーネント（Heat）とそれ以外の差異(2)

- 管理 – Heatと他は何が異なるのか？
  - **Watcher**
    - Heatは基本的に資源の作成を目的とするが、Watcherは作成済み資源の配置の最適化を目的とする。
    - 最適化の元となるデータの収集にはCeilometerやMonascaを利用する。
    - CeilometerやMonasca（またはCongress？）で監視を行い、最適化操作はMistralにやらせれば良いのでは？という気もするが、Watcherは最適化戦略を作り込んでおり、その点が強みとなっているようだ。

ところで.....

各々違いがあることは分かった！  
でも全部HeatやCeilometerの新機能として実装すればよくない？

そうかもしれませんが.....

各々違いがあることは分かった！  
でも全部HeatやCeilometerの新機能として実装すればよくない？

- そもそも出自が分かれている。
- 現実問題として、
  - 1つのコンポーネントにあれもこれも詰め込むと肥大化しすぎる。
  - 既存コンポーネントには既に機能や実装に対するポリシーがあり、大きな新機能だと調整にコストがかかりすぎる問題もある。
- また、「Big Tent」により、プロジェクトの追加について以前より柔軟に認めていく体制になったことも関連。

## 監視・管理コンポーネントのBlueprints紹介

---

これまでご紹介したコンポーネントで現在進行中のBlueprintsをいくつか紹介します。

- 実用上、幅広いシチュエーションで役立ちそうなBP
- 技術的に面白みがあるBP

# 注目Blueprints - Mistral

- Newtonで実装される（かもしれない）注目Blueprint  
※リリースターゲットがNewtonとなっているものから抽出。
- **Multi region support**  
<https://blueprints.launchpad.net/mistral/+spec/mistral-multi-region-support>
  - Mistralが現状マルチリージョン環境に対応できていないため、対応させるBP。
  - Mistralで作業を自動化したい場合、ある程度以上大規模、もしくは複雑な操作が要求される環境が予想されるので、本機能の実装は重要と思われる。

## 注目Blueprints - Congress

- Newtonで実装される（かもしれない）注目Blueprint  
※リリースターゲットがNewtonとなっているものから抽出。
  - **Push Type Datasource Driver**  
<https://blueprints.launchpad.net/congress/+spec/push-type-datasource-driver>
    - これまでCongressはPull式に情報を収集していたが（e.g. nova APIの定期実行）、本BPが実装されるとPush式に情報を受け取れるようになる。
    - Congress自体の改造なしに情報収集対象を広げられるため、より広範な利用が可能になると思われる。

## 注目Blueprints - Monasca

- 今後実装される（かもしれない）注目Blueprints  
※プロジェクトごとの文化の違いか、MonascaのBlueprintは大半がリリースターゲットを定めないまま進められている。

- **Alarms on Logs**

<https://blueprints.launchpad.net/monasca/+spec/alarmonlogs>

- Monascaでログ監視を行うためのBP。  
指定した以上のログレベルが検出された場合にアラームを発信するなどの使い方が想定されている。
- Monascaが監視可能な対象は現状で既にかかなり豊富なのだが、ログは穴となっていたのでこれが実装されるとより高度な監視が可能になると思われる。

## まとめ、およびこれからの展望

- OpenStackは仮想資源の単純なCRUDだけではない、**監視・管理**といった運用面も含めた**包括的なサービス**を提供できるようになってきた。
- ただし、これまで挙げたようなコンポーネントをなんでもかんでも導入してしまうと、**導入・運用コストのほうが高くつくことも有り得る**。  
当たり前だが、システムごとに現在の運用で不足している/コストが高い箇所を見極めたうえで導入していくべき。
- 完全に独断ですが、適用範囲の幅広さや機能の柔軟性から、特に**Congress・Mistral・Monasca**は今後人気になるかもしれない。

---

ご清聴ありがとうございました。