

# 社内プライベートクラウド品質向上に向けた デプロイ, テスト, ベンチマークの自動化

---

NTTコミュニケーションズ株式会社

技術開発部

**松本 赳明**

[takeaki.mastumoto@ntt.com](mailto:takeaki.mastumoto@ntt.com)

NTTコミュニケーションズ  
技術開発部  
エンジニア

2015~  
OpenStack研究開発

Tempest等のコンポーネントにコントリビュート



**Global ICT Partner**  
Innovative. Reliable. Seamless.

# Agenda

---

1. 会社概要
2. 背景
3. デプロイとベンチマークの自動化
  - デプロイ (Cobbler, Chef)
  - テスト (Tempest)
  - ベンチマーク (Rally)
  - 評価
4. ユースケース
5. まとめ
6. Rally最新動向

## OpenStackプロダクト

- Enterprise Cloud 2.0
- Cloud<sup>n</sup> VPC type ClosedNW

## OpenStackコミュニティ活動

- OpenStack Foundation Corporate Sponsor
- Japan OpenStack User Group
- Bug Report / Bug Fix / Blueprint

### 目的

- 運用ノウハウの蓄積（バグフィックス、アップデート...）
- 新しいコンポーネントや機能の検証
  - ⇒頻繁に設定や構成を変更、壊してもすぐに戻せる必要

### OpenStackの構築

- 大量のコンポーネント、モジュール、設定ファイル
- 複雑なネットワーク構成
- 冗長化
  - ⇒OpenStackのデプロイは容易にできない

## Our Challenge

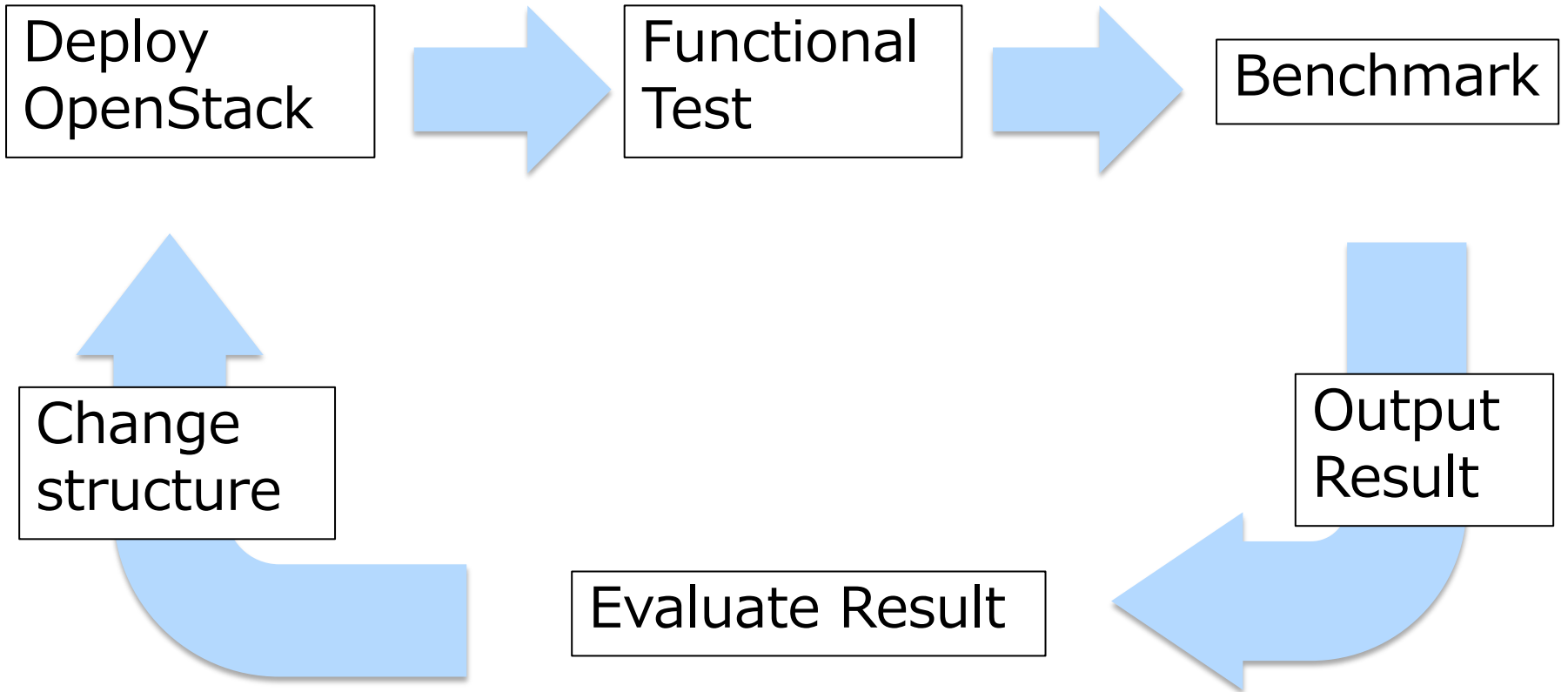
---

どうやったら品質を維持したまま、継続的なデプロイができるか？

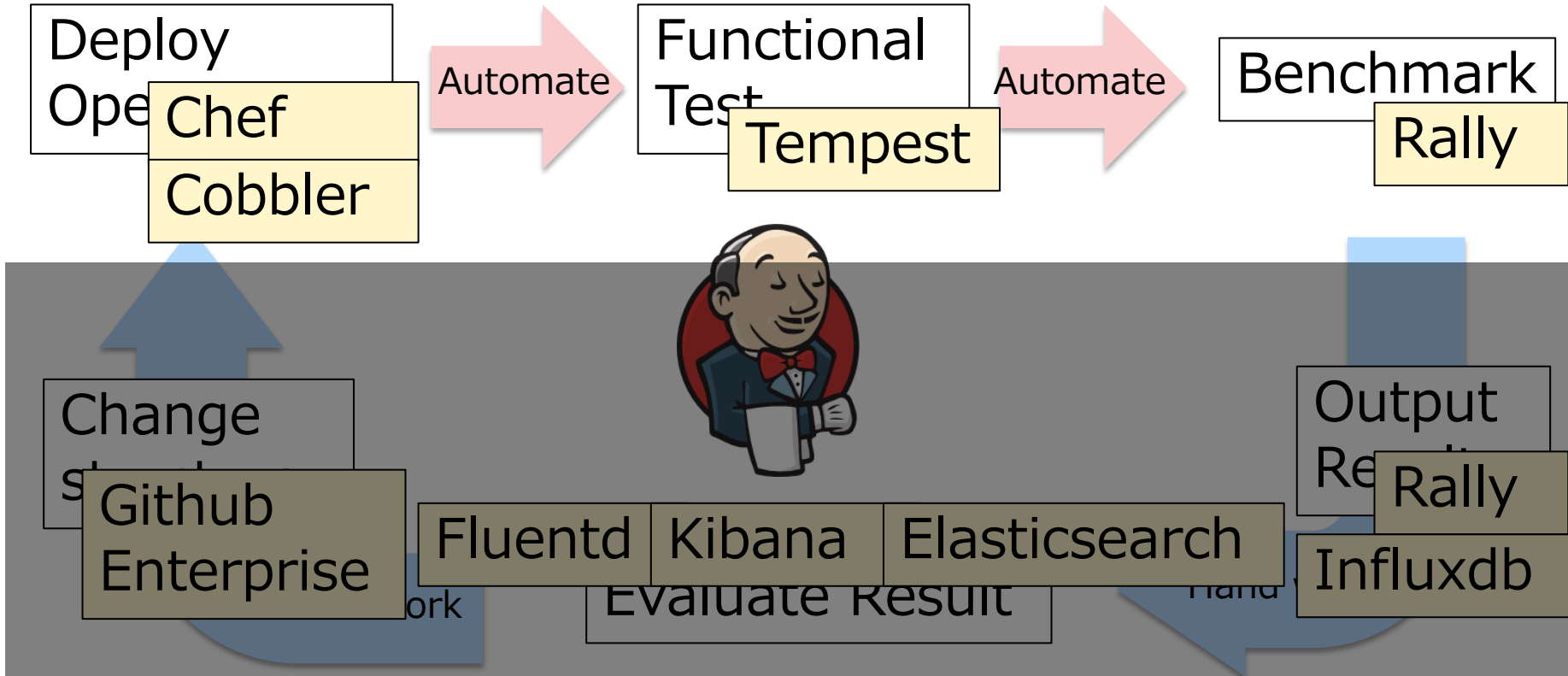
## 2. デプロイとベンチマークの自動化

---

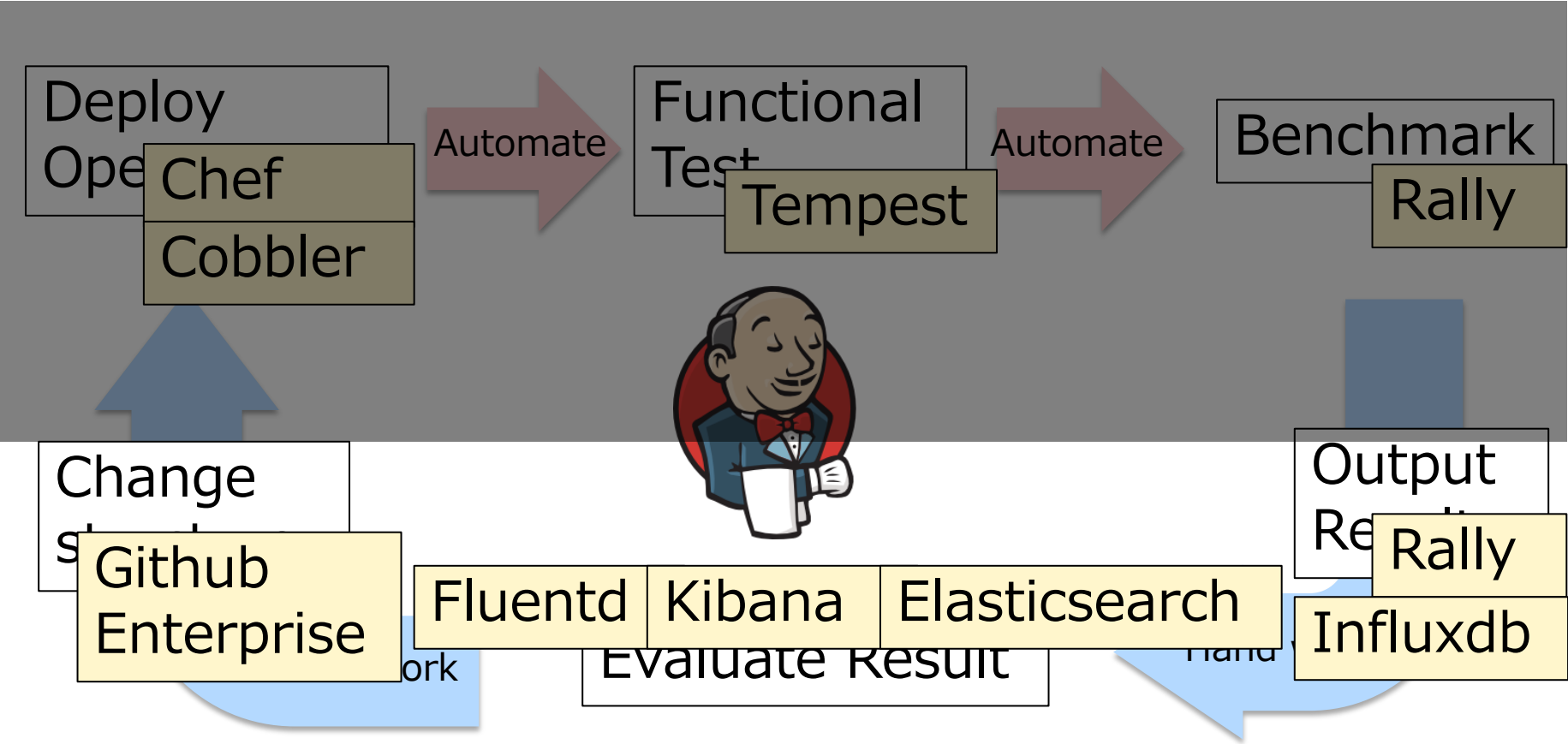
# Improvement cycle



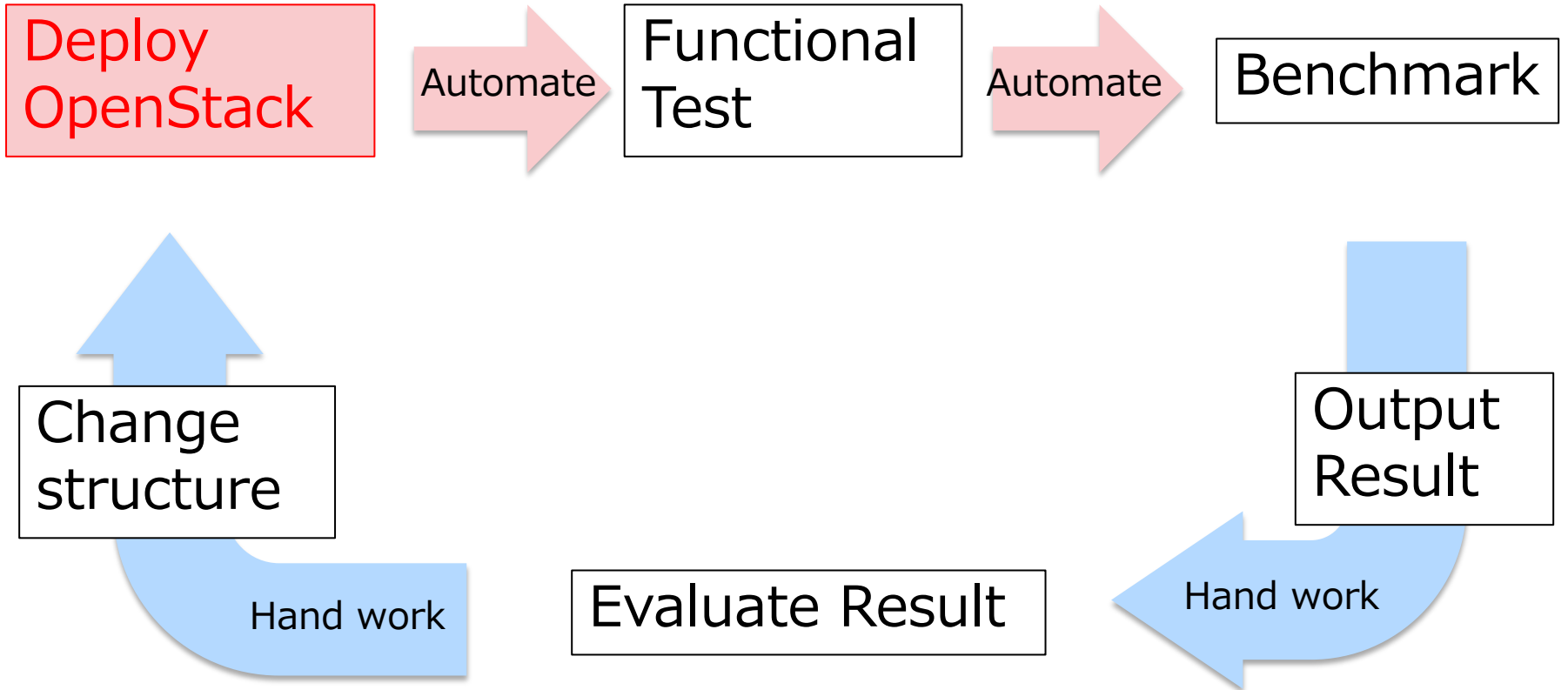
# Our Implementation (2/2)



# Our Implementation (2/2)

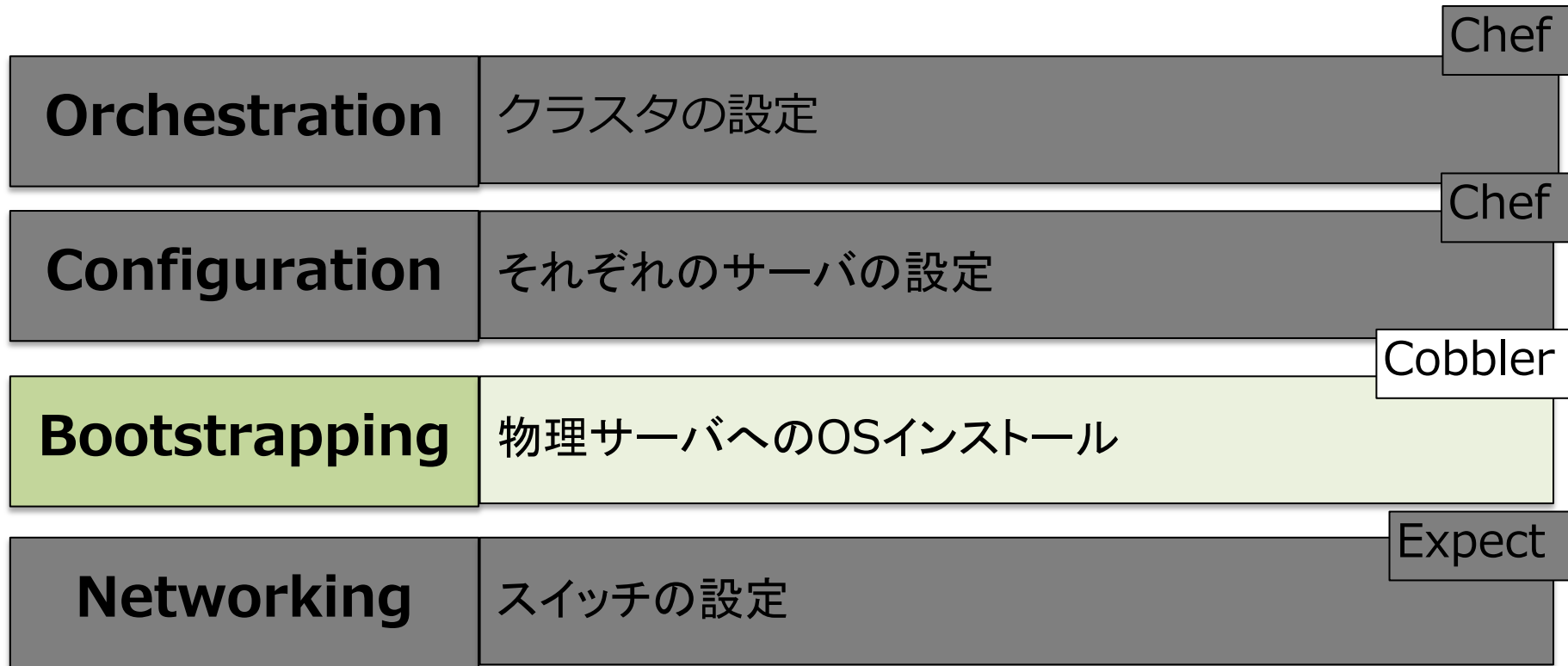


# Deployment







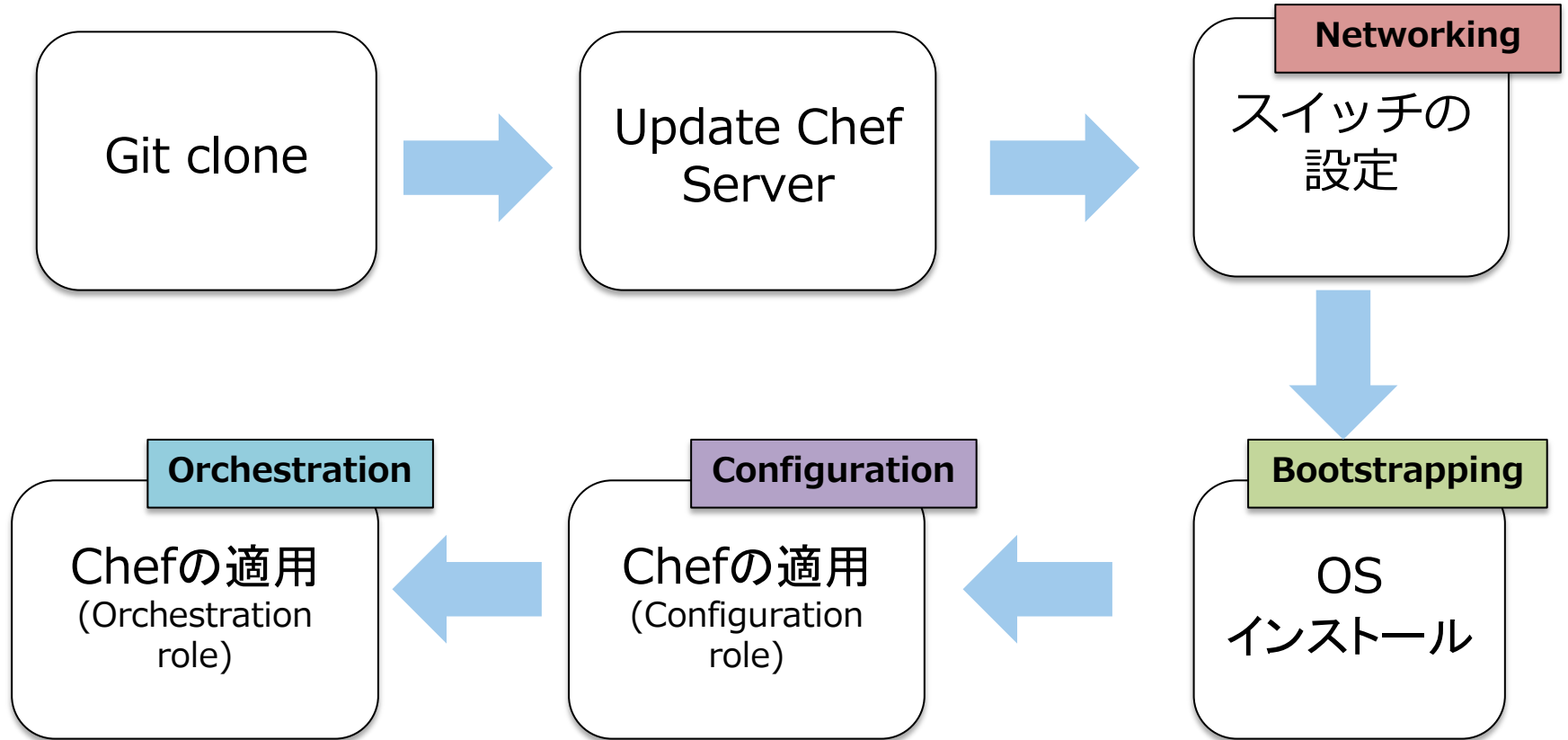


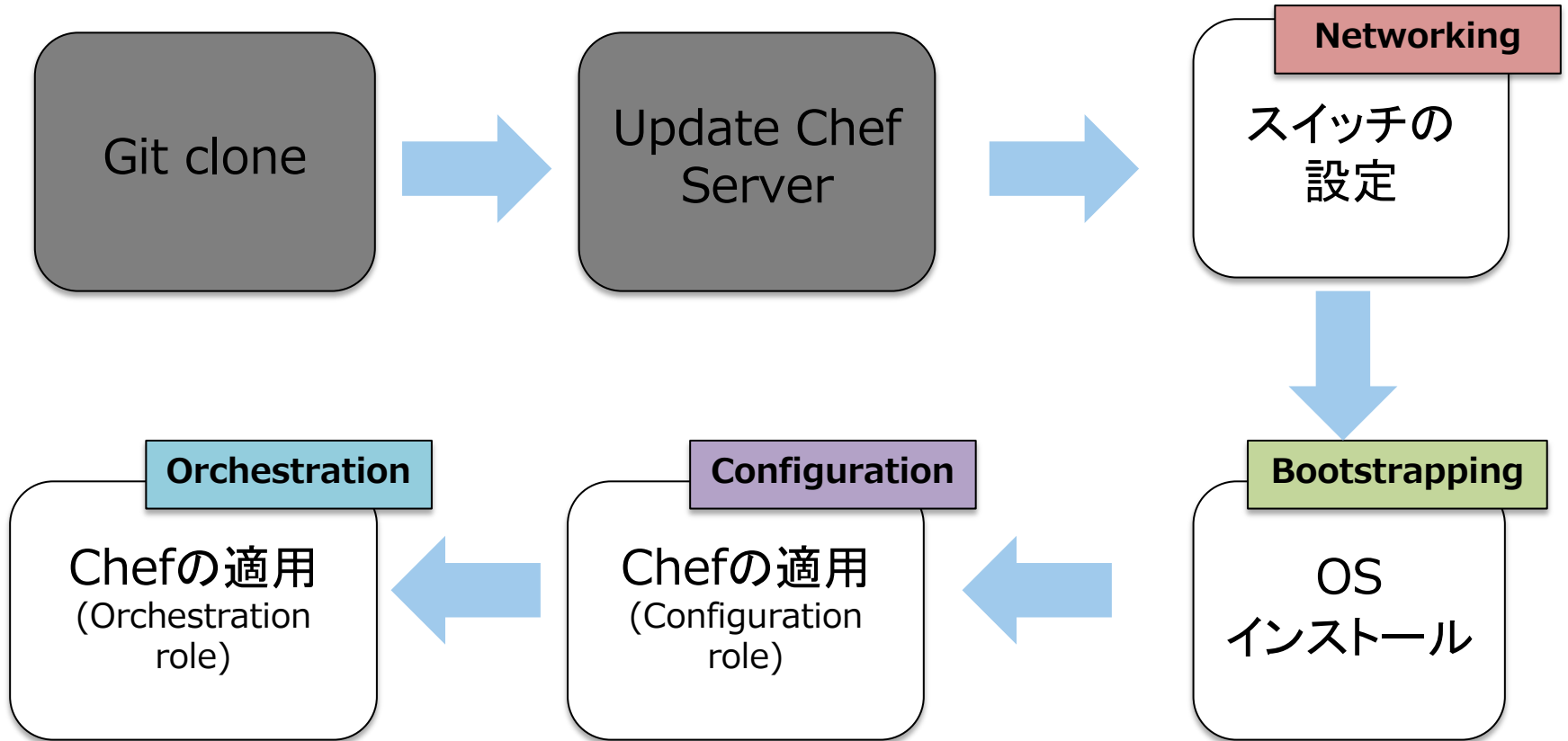


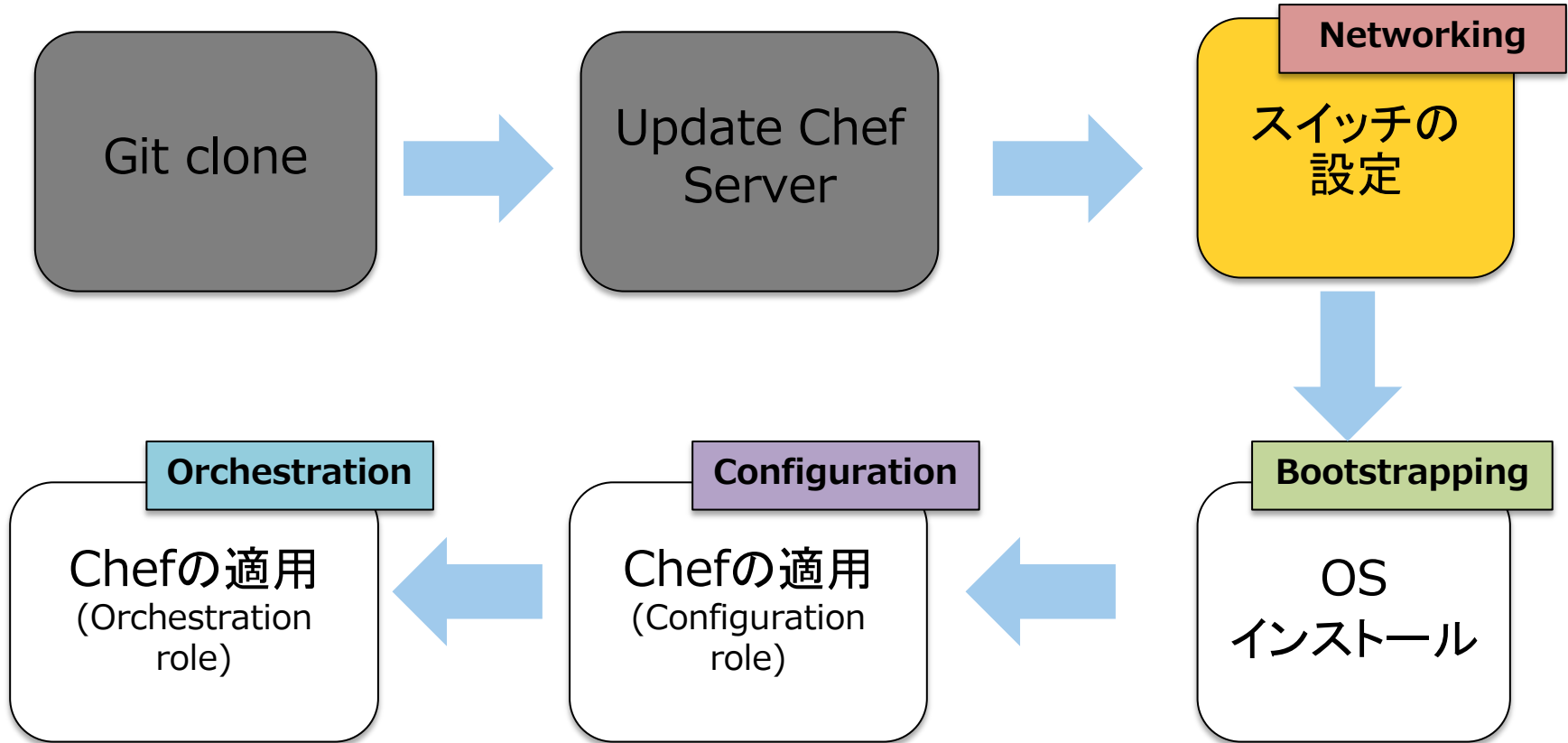


## 環境情報はすべてテキストフォーマットで記述し、Gitでバージョン管理する

- ネットワークの設定
- OpenStackの構成
  - ✓ どのサーバがどの役割をもつノードか
- それぞれのノードの設定
  - ✓ Chef Cookbook, Repository





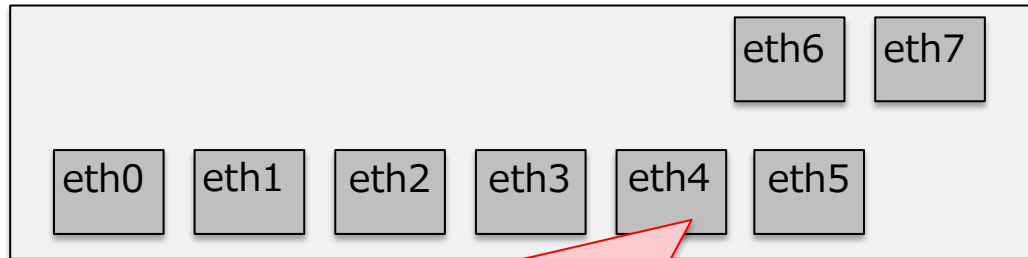


- VLANの自動設定
- 必要条件
  - ✓ server\_port\_nameの定義
  - ✓ 命名規則に沿ったスイッチのinterface descriptionの設定  
" <server\_name>\_<server\_port\_name> "

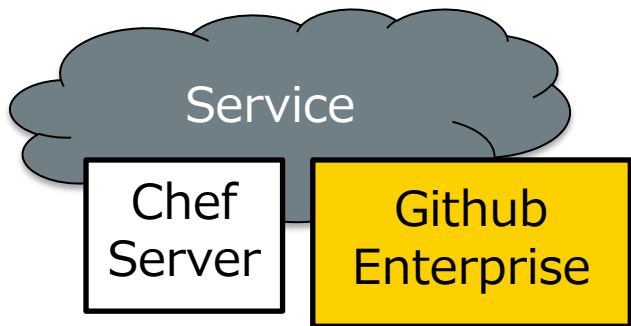
#### Config.yaml

```
- server_1:  
  - eth0:  
    - vlan10:  
      - native  
    - vlan100  
  - eth1:  
    - vlan10:  
      - native
```

#### Naming rule for physical port



When I got servers, I decide it

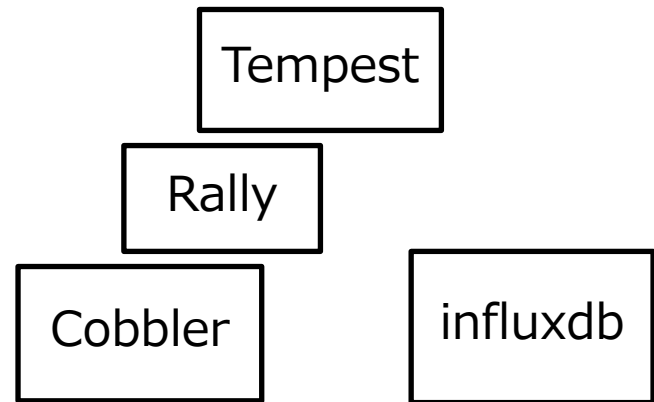


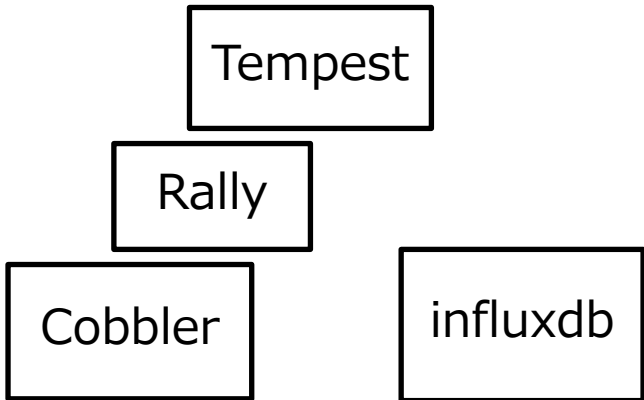
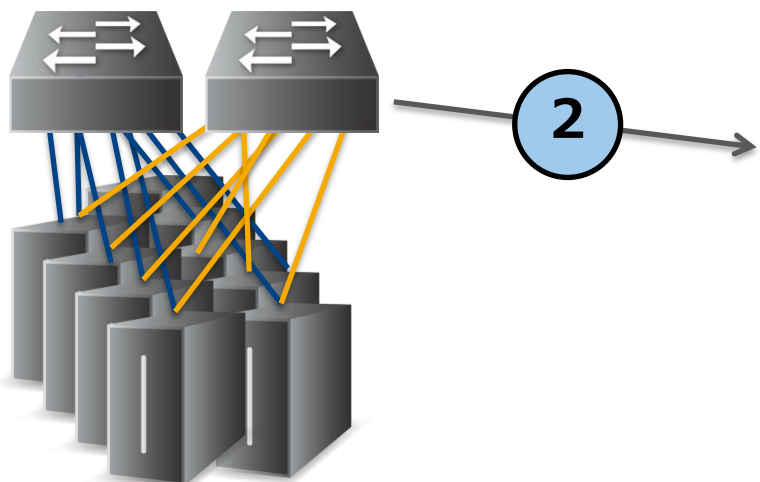
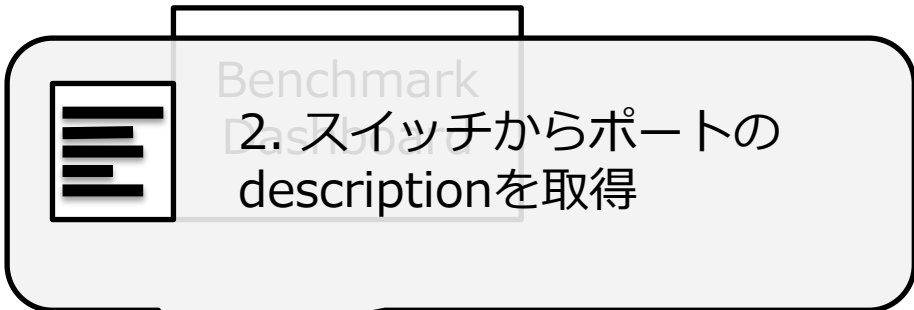
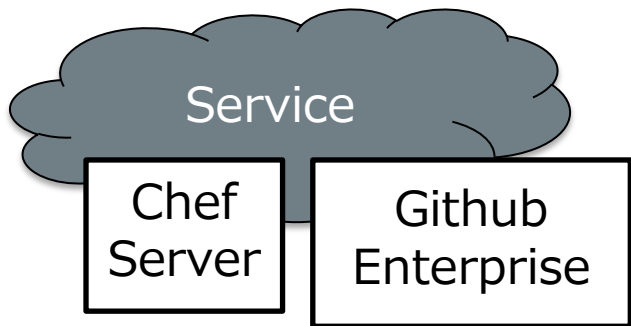
config.yaml (image)

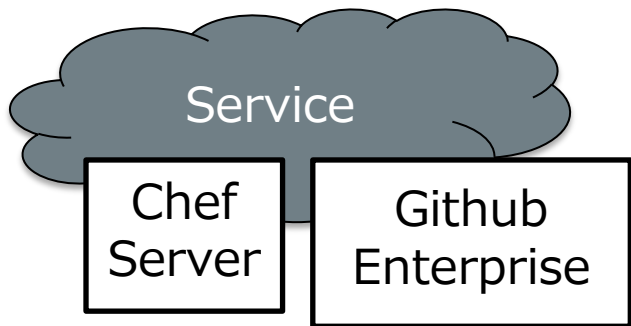
- server\_1
  - eth0
    - vlan10
    - native
    - vlan100
  - eth1
    - vlan10

1

1. VLAN設定を含む設定の取得

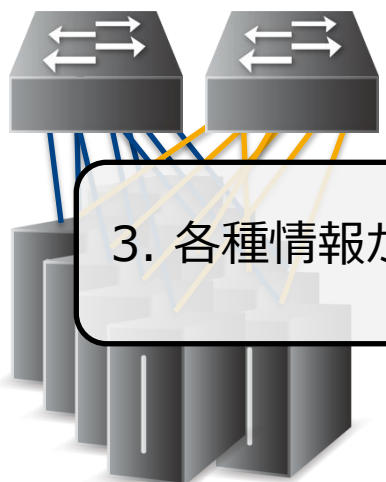




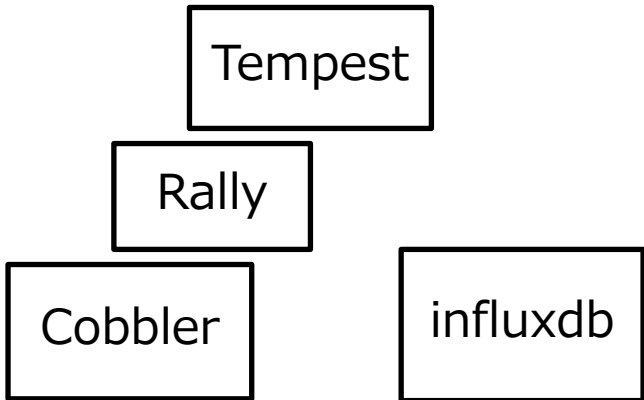


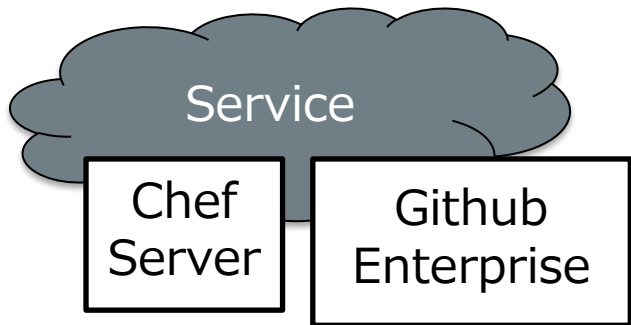
3

```
Server1 eth0(switch1 port 1/15)  
=> VLAN100, 200  
Server1 eth1(switch2 port 1/15)  
=> VLAN100, 200
```

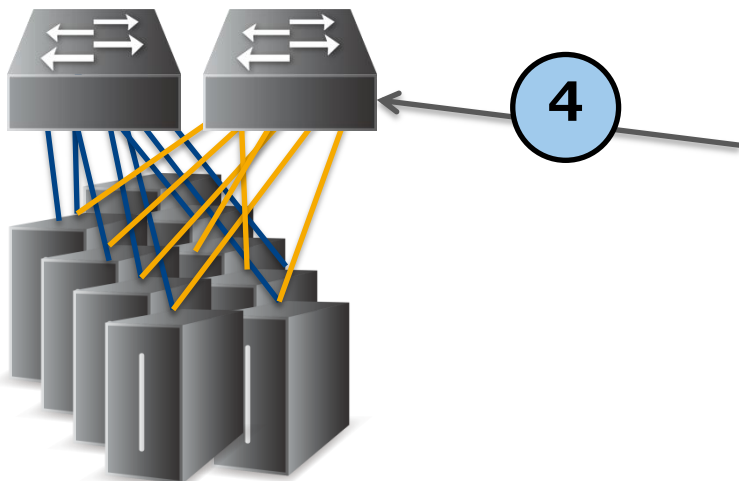


3. 各種情報からスイッチの設定を生成





4. 生成されたコマンドの実行



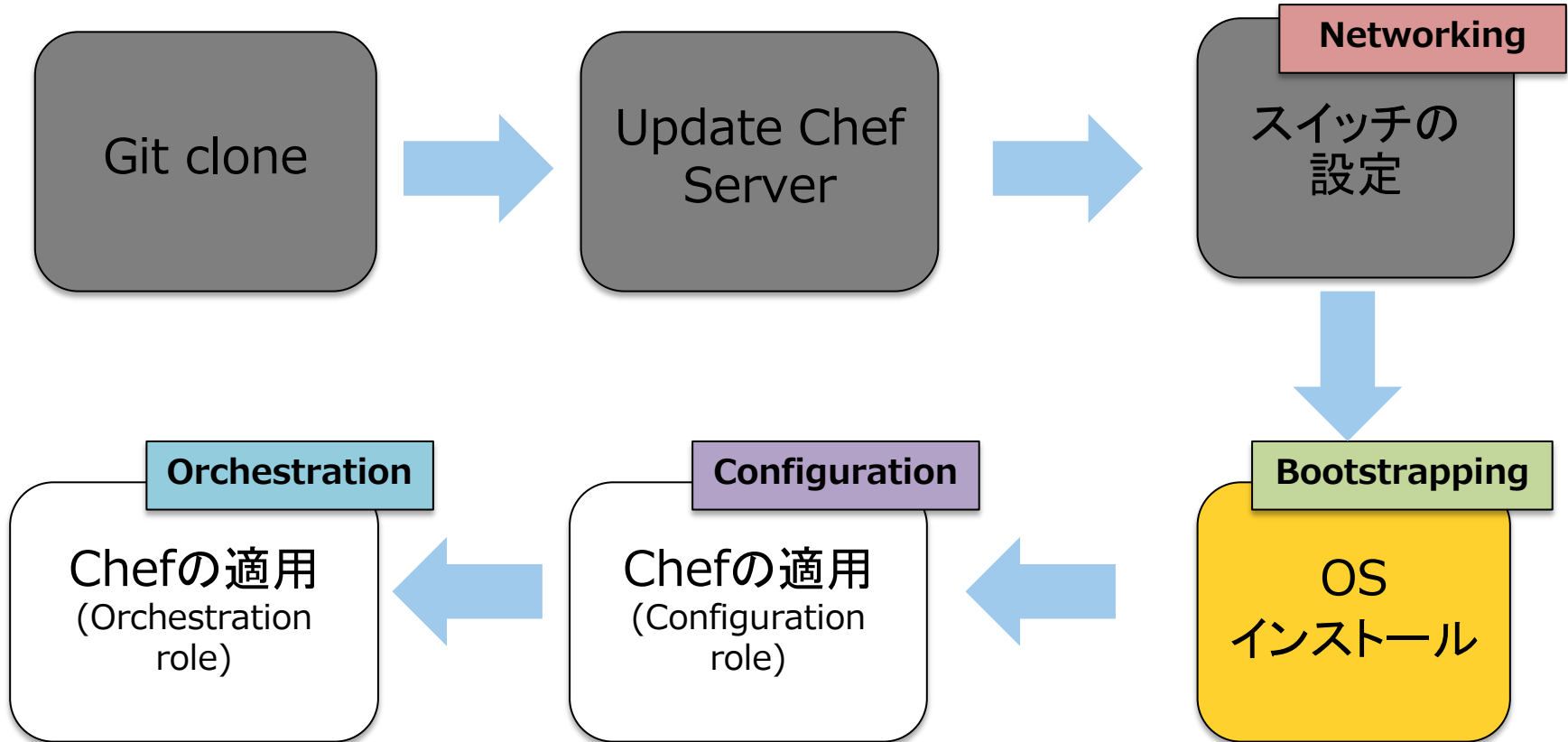
Benchmark Dashboard

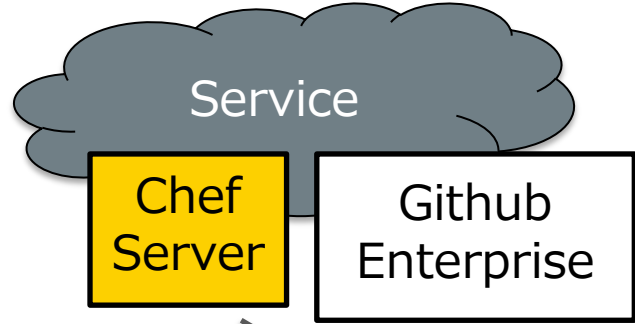
Tempest

Rally

Cobbler

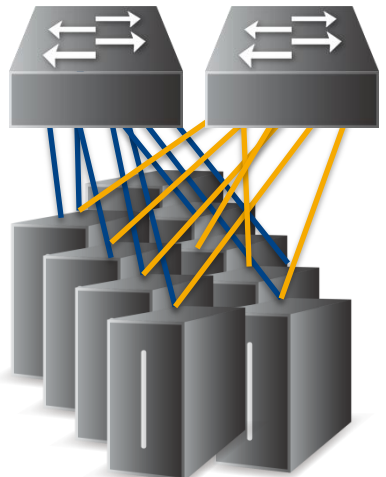
influxdb





1. Chefの適用  
2. Cobblerのインストールと設定

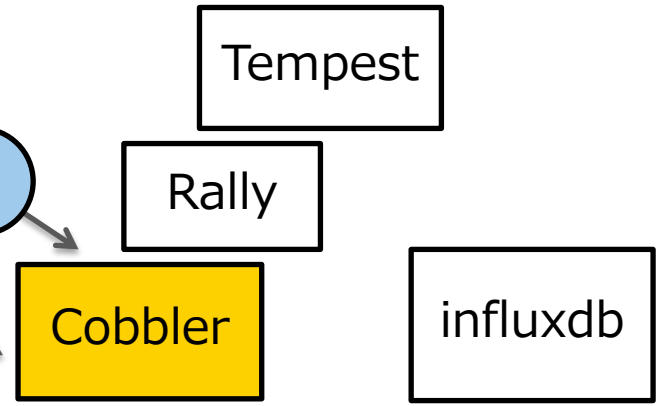
- ✓ サーバの登録
- ✓ Hostname
- ✓ IP address

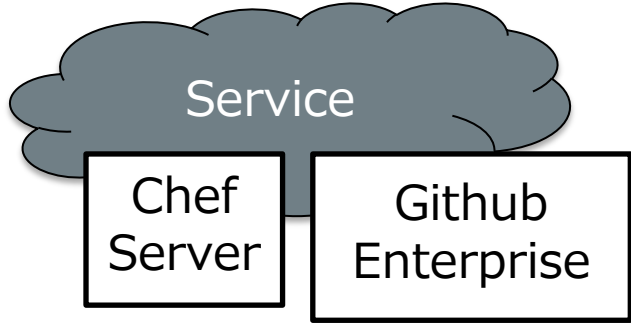


2



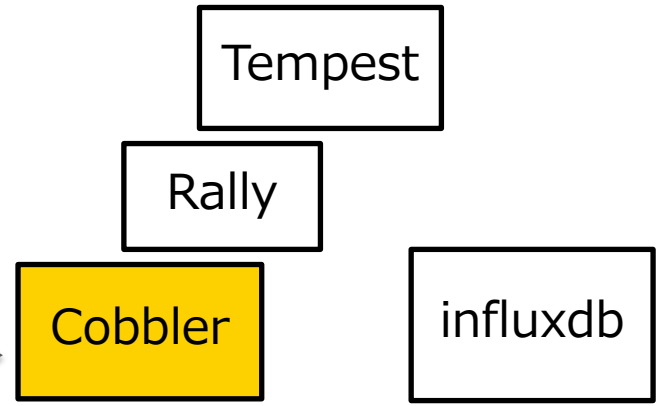
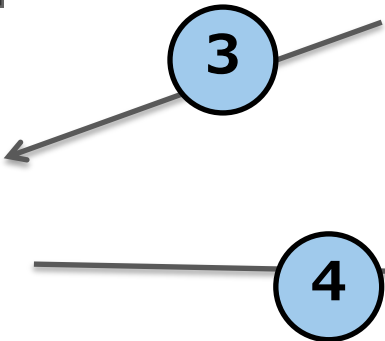
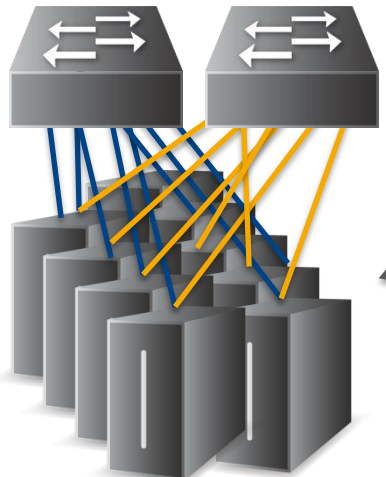
1

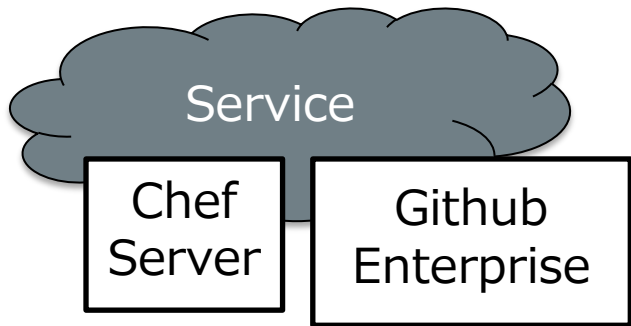




Benchmark  
Deployment

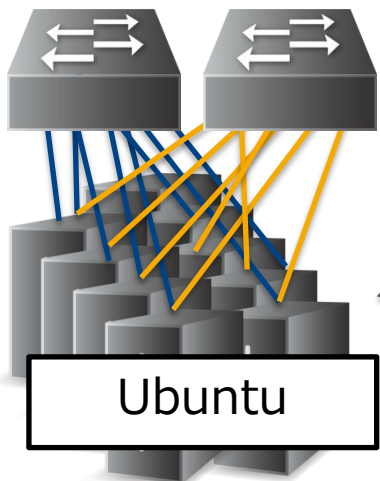
3. IPMI power reset  
4. PXEで自動インストールが開始



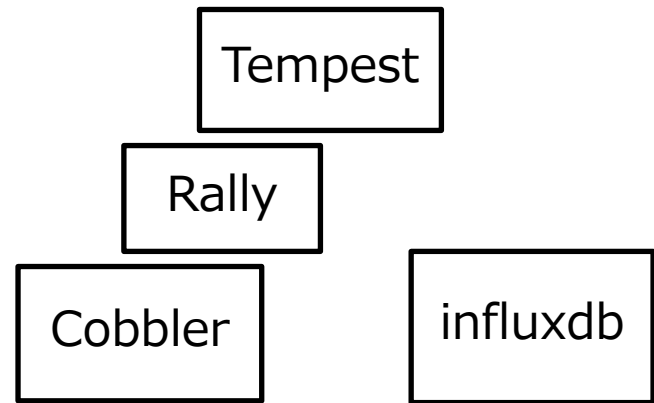


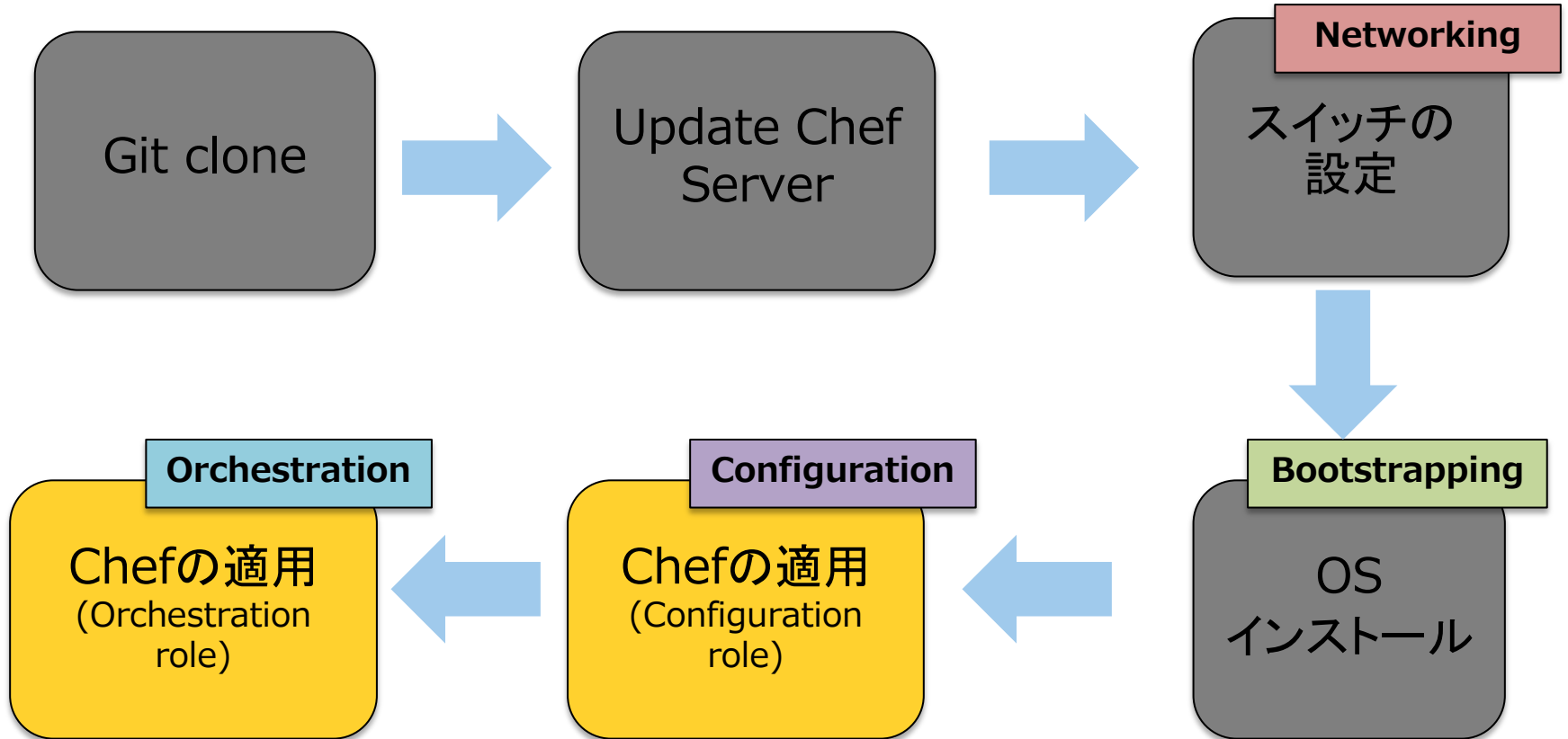
Benchmark Dashboard

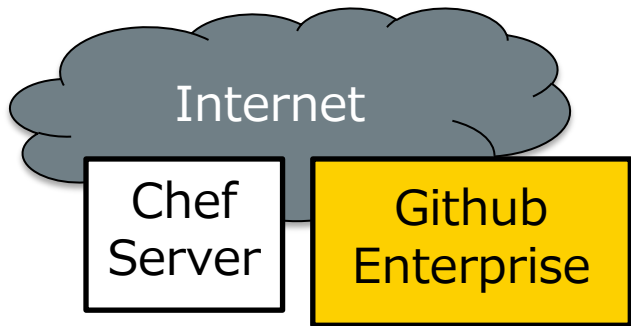
5. SSHで接続できるかチェック



5







1

1. Git cloneでOpenStackの構成情報を取得

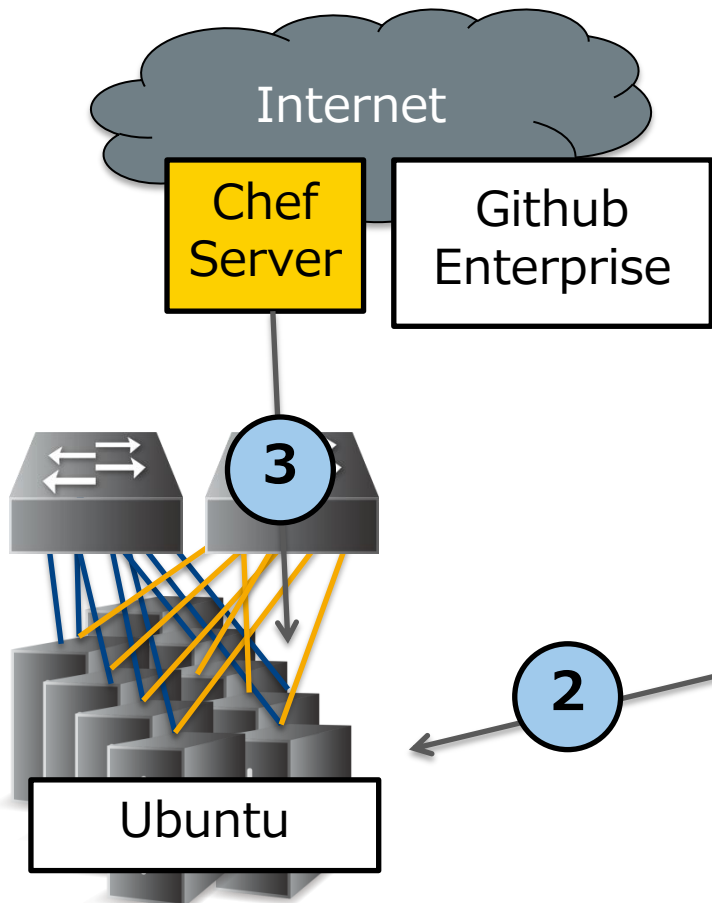
Ubuntu

definition.json

```
{
  "<environment>": {
    "servers": {
      "server1": "backend",
      "server2": "controller"
    },
    "dependencies": {
      "backend": [],
      "controller": [{"backend": 1}]
    }
  }
}
```

Cobbler

influxdb



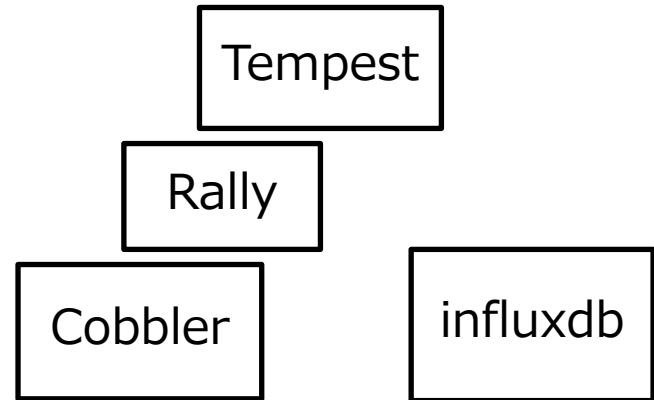
Benchmark Dashboard

2. Chefの適用  
3. roleの適用

- ✓ backend configuration role
- ✓ backend orchestration role...



2





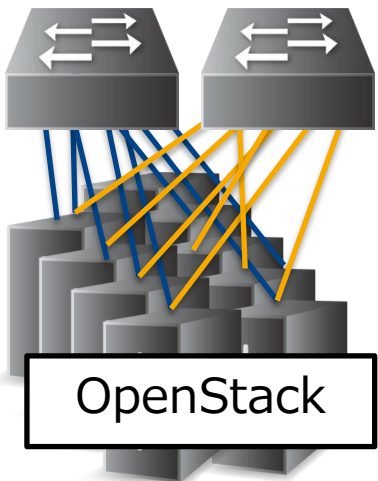
Chef Server

Github Enterprise

Benchmark Dashboard

4. Orchestration roleの削除  
 ✓Orchestration roleは幕等性がないため

4

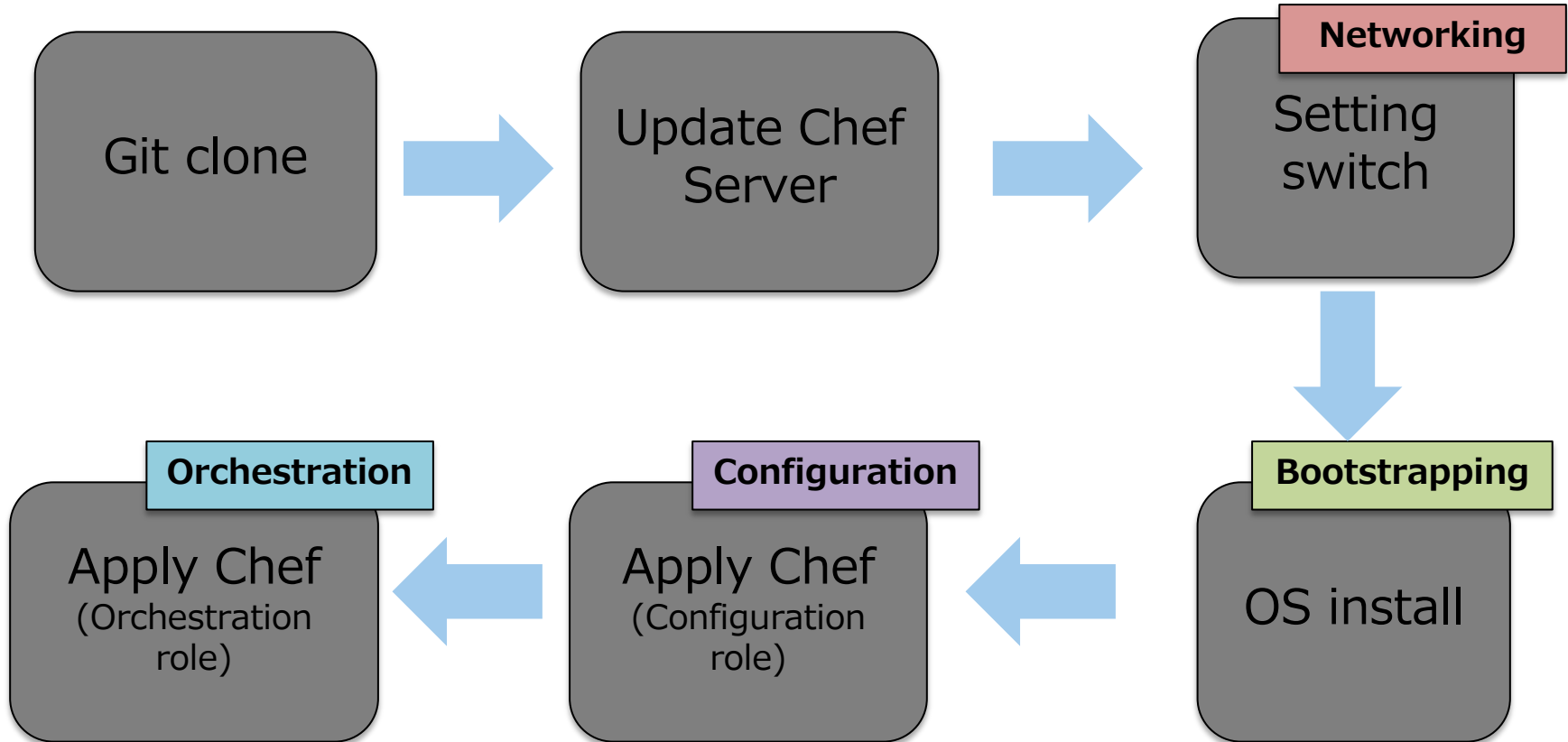


Tempest

Rally

Cobbler

influxdb



## Pros

- いつでも誰でも同じ環境を構築することができる

## Cons

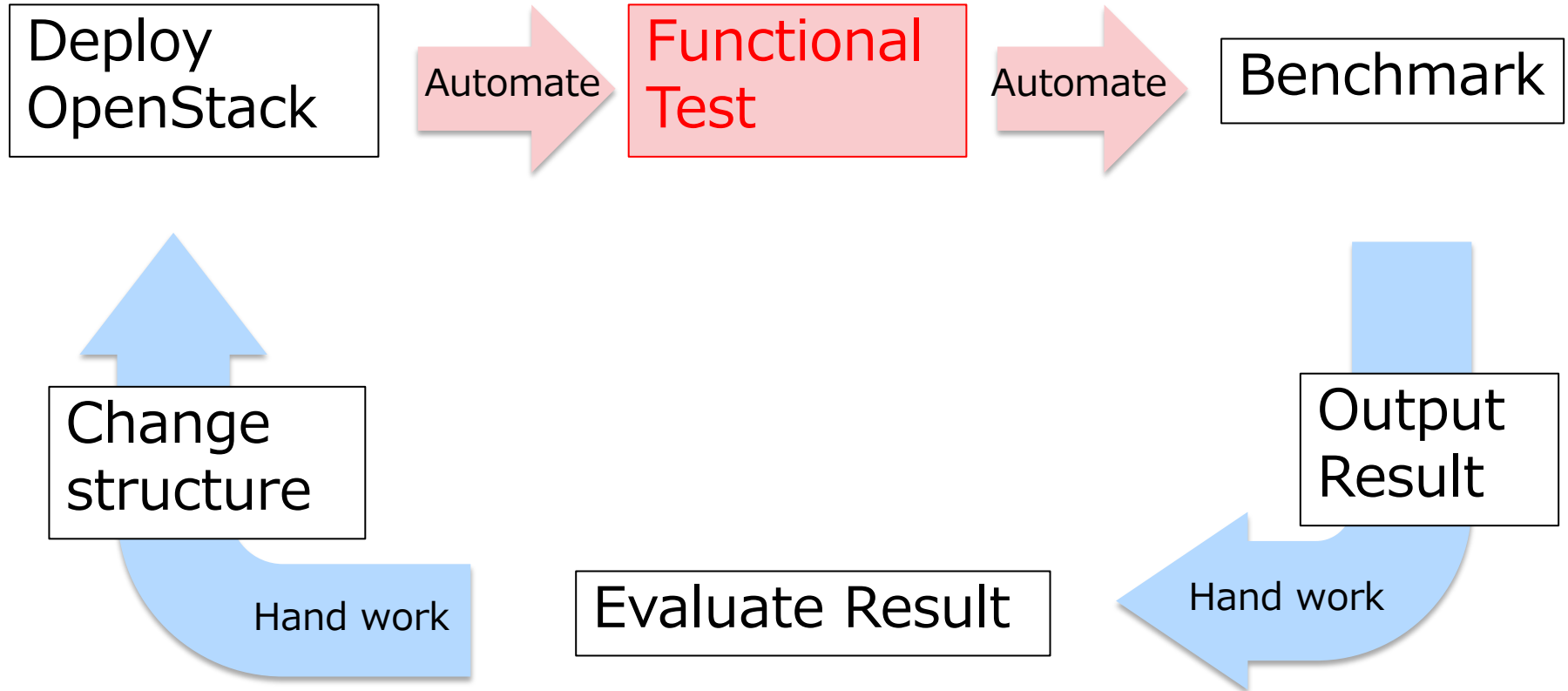
- 複数のツールを複雑に組み合わせる必要がある
  - ✓ 広い知見が必要

# 機能テスト

---

# Functional Test

※ Omit detail process here



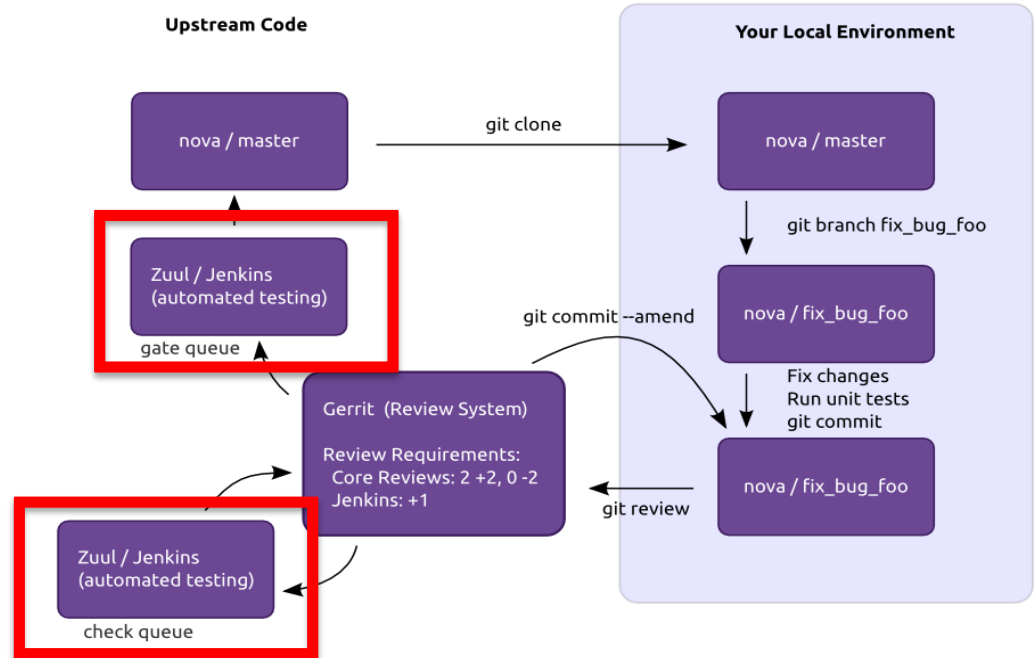
## テストに使用したツール

- Tempest

## Tempestとは?

- Tempestは公式のインテグレーションテストスイート
- “Tempest can test any OpenStack, all in one or 1000 nodes”

## Tempestはupstreamの開発に使用されている



Community Workflow by <http://docs.openstack.org/>

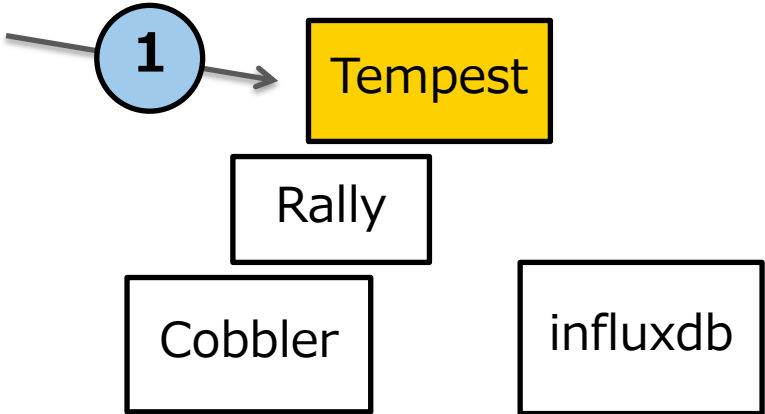
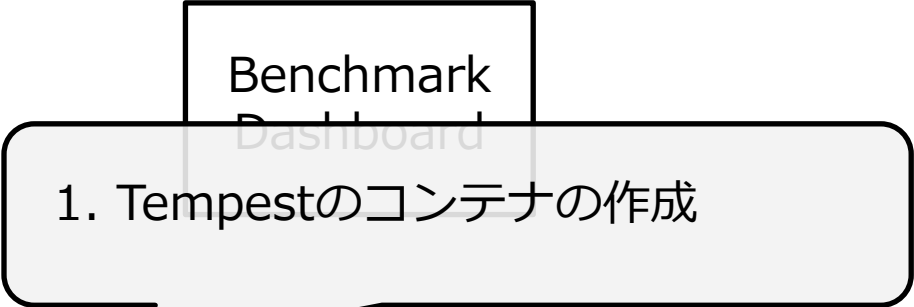
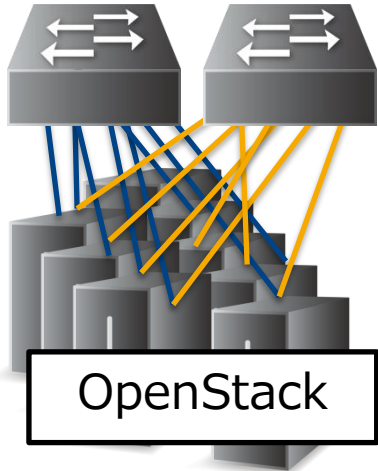
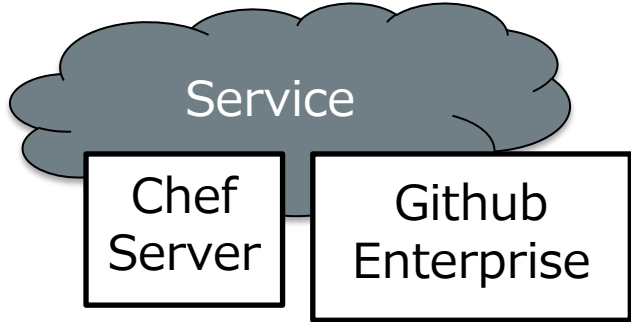
### 実行しているテスト

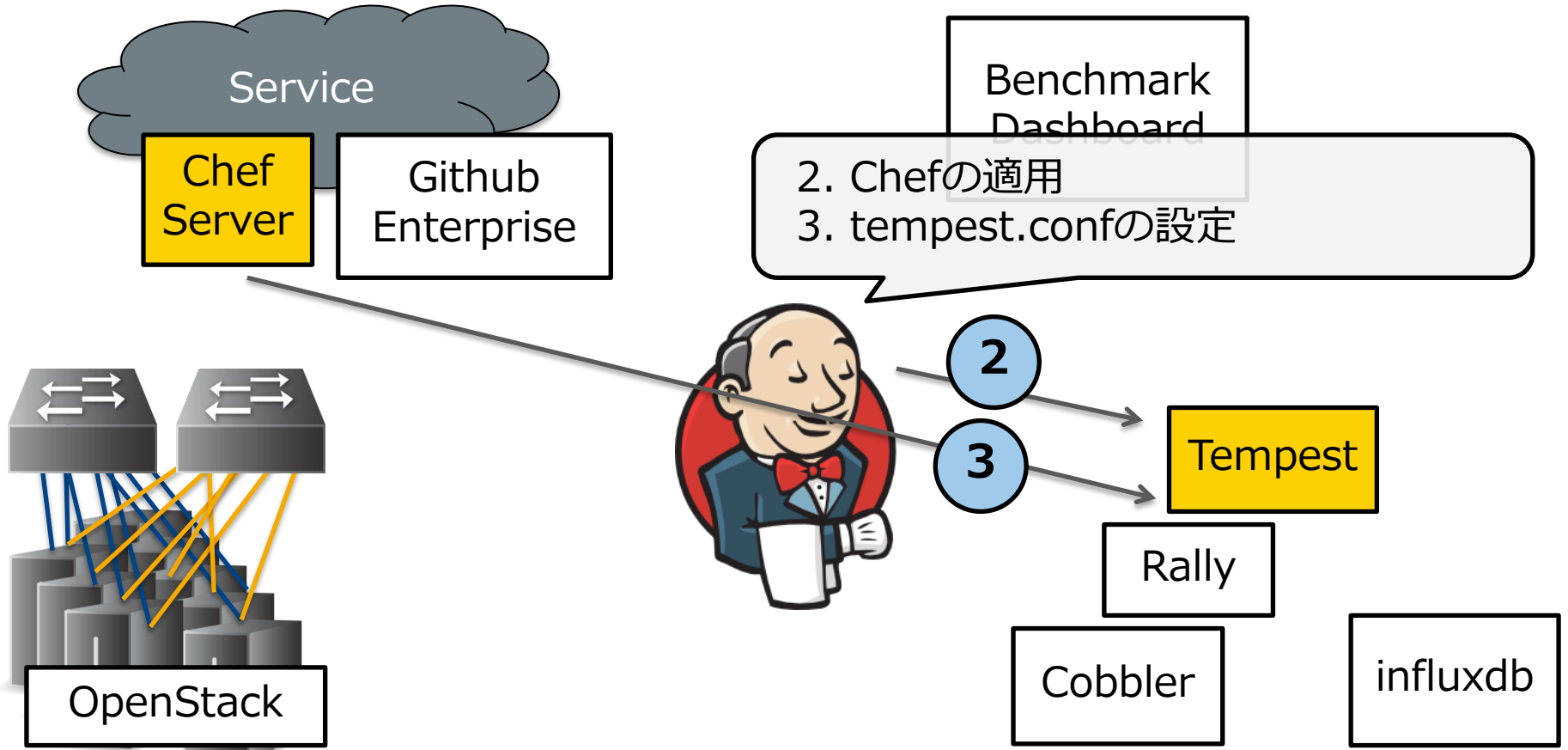
- APIテスト
  - ✓ それぞれのAPIが単体で正しく動くかのテスト
- シナリオテスト
  - ✓ ユーザーが実際行うような一連の操作のテスト

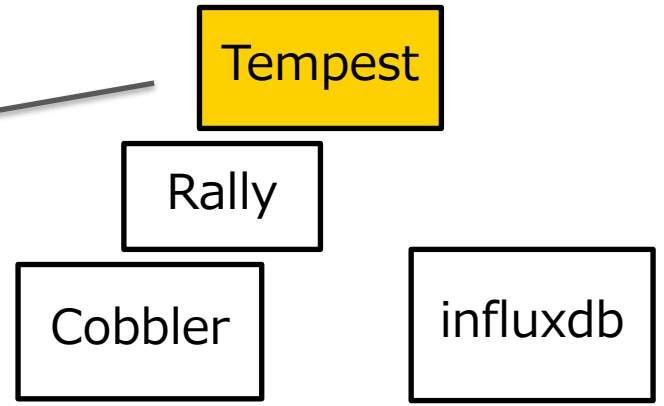
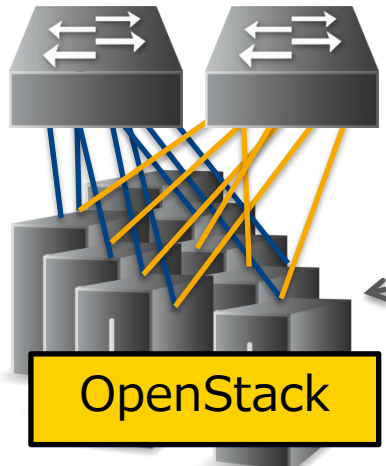
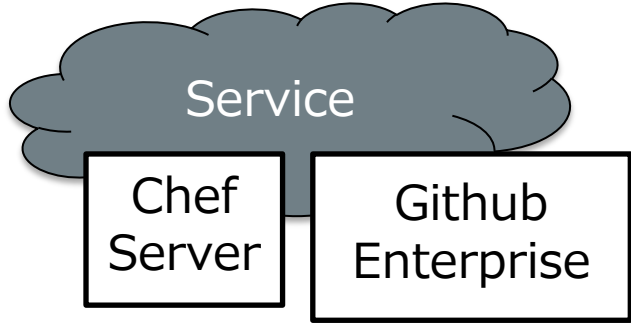
## 実行方法

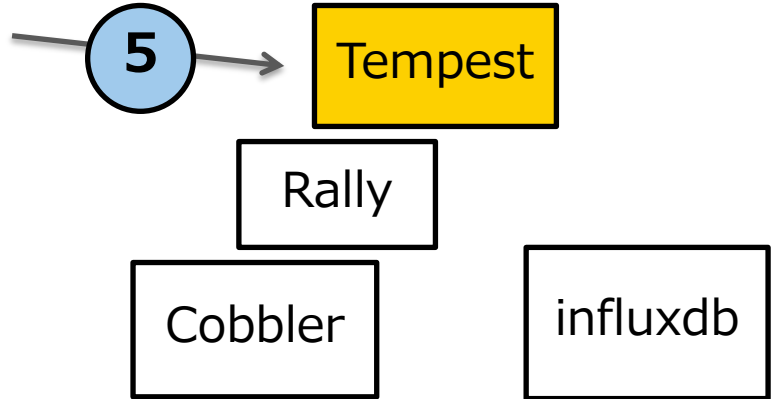
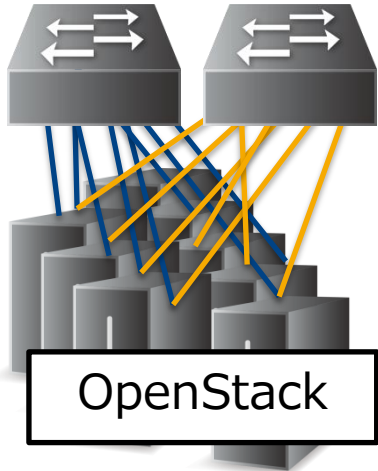
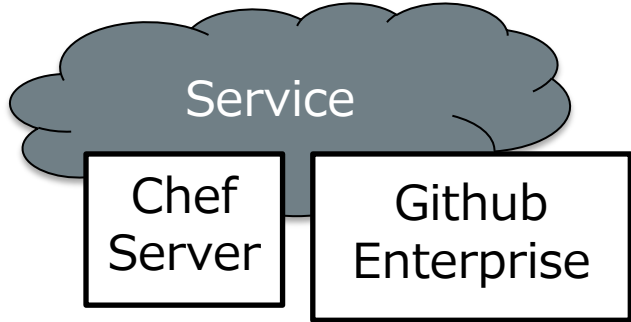
1. tempest.confの編集
2. ./run\_tempest.sh の実行 (ostestr, 将来的にはtempest)

```
$ ./run_tempest.sh
.....
Ran 1124 tests in 1185.267s
FAILED (failures=4)
```









## Pros

- 環境に応じた設定の自動デプロイ
- カスタマイズしたシナリオの実行ができる

## Cons

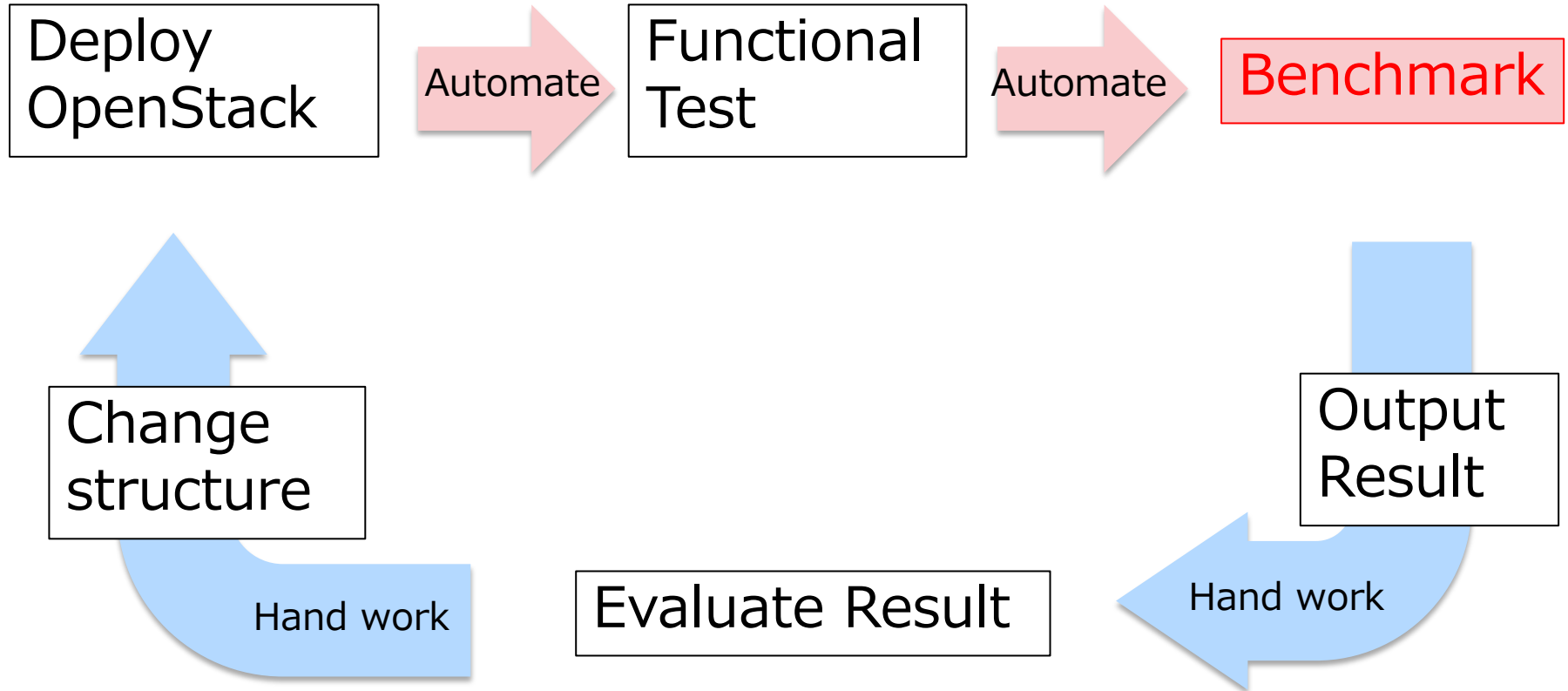
- シナリオがシステムの構成パターンを網羅していない
  - ✓ Tempestに11個のパッチを提出
- APIレベルでテストケースの制御ができない

# ベンチマーク

---

# Benchmark

※ Omit detail process here



## ベンチマークツール

- Rally ( is one of the OpenStack projects )

## Rallyとは？

- Released v0.4.0 in 18 Apr 2016
- ベンチマークツール
  - ✓ デプロイやテストもサポートしている

# 1. Deploymentの作成

```
rally@3eba20593250:~$ cat config.json
```

```
{  
  "admin": {  
    "password": "p@ssw@rd",  
    "tenant_name": "admin",  
    "username": "admin"  
  },  
  "auth_url": "http://lb:5000/v2.0",  
  "type": "ExistingCloud"  
}
```

既に構築済みのOpenStackを登録

```
rally@3eba20593250:~$ rally deployment create --filename config.json --name existing
```

## 2. Credentialの確認

```
rally@3eba20593250:~$ rally deployment check
keystone endpoints are valid and following services are available:
```

```
+-----+-----+-----+
| services | type   | status |
+-----+-----+-----+
| cinderv2 | volumev2 | Available |
| glance  | image   | Available |
| keystone | identity | Available |
| neutron | network | Available |
| nova    | compute | Available |
+-----+-----+-----+
```

### 3. task fileの作成

```
rally@3eba20593250:~$ cat benchmarks/nova/boot-and-delete.json
{% set flavor_name = flavor_name or "m1.tiny" %}
{
  "NovaServers.boot_and_delete_server": [
    {
      "args": {
        "flavor": {
          "name": "{{flavor_name}}"
        },
        "image": {
          "name": "^cirros.*$"
        }
      }
    }
  ]
}
```

Taskファイルはjsonかyamlで記述できて、  
Jinja2の文法が使える。

## 4. ベンチマークの実行

```
rally@3eba20593250:~$ rally task start benchmarks/nova/boot-and-delete.json
```

Response Times (sec)								
action	min	median	90%ile	95%ile	max	avg	success	count
nova.boot_server	4.614	4.768	4.972	5.417	5.861	4.852	100.0%	10
nova.delete_server	2.486	2.576	2.709	2.732	2.754	2.603	100.0%	10
total	7.148	7.369	7.658	8.018	8.377	7.455	100.0%	10

```
Load duration: 37.9388210773
```

```
Full duration: 51.1114509106
```

```
...
```

## 5. 結果をHTMLで出力

```
rally@3eba20593250:~$ rally task report --tasks 5e8815a9-10eb-4d67-86b4-6027f4adc634  
1f71cb68-9c1e-49e3-8af5-6d21dc34c455 --out result.html --html
```

Rally benchmark results

**Benchmark overview**

Input file

- ▶ KeystoneBasic
- ▶ NovaHypervisors
- ▶ NovaKeypair
- ▶ NovaSecGroup
- ▼ **NovaServers**
- boot\_and\_delete\_multiple\_servers
- boot\_and\_delete\_server**
- boot\_and\_list\_server
- boot\_server
- list\_servers
- shelve\_and\_unshelve\_server
- snapshot\_server

## NovaServers.boot\_and\_delete\_server (51.609s)

Overview Details Input task

Load duration: **38.166 s** Full duration: **51.609 s** Iterations: **10** Failures: **0**

### Total durations

Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)
nova.boot_server	4.645	4.869	5.53	2.963	5.926
nova.delete_server	2.476	2.523	2.679	1.424	2.849
<b>total</b>	<b>7.267</b>	<b>7.377</b>	<b>8.096</b>	<b>4.217</b>	<b>8.775</b>

### Charts for the Total durations

● Stacked ○ Stream ○ Expanded

● duration ● idle\_duration

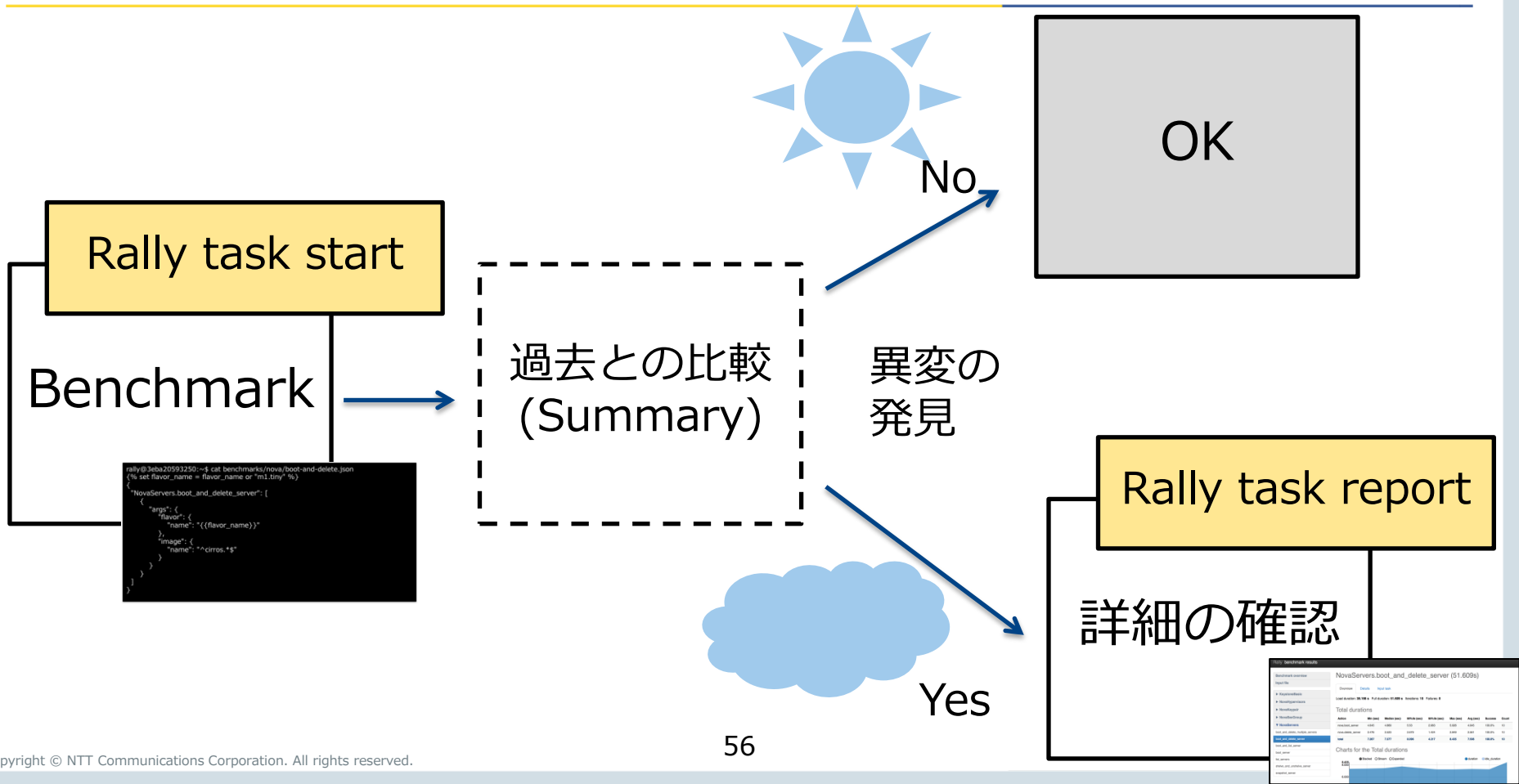
ベンチマークの種類ごとに詳細結果を確認できる。

※1 There is blueprint

<https://blueprints.launchpad.net/rally/+spec/cli-plot2html-compare-tasks>

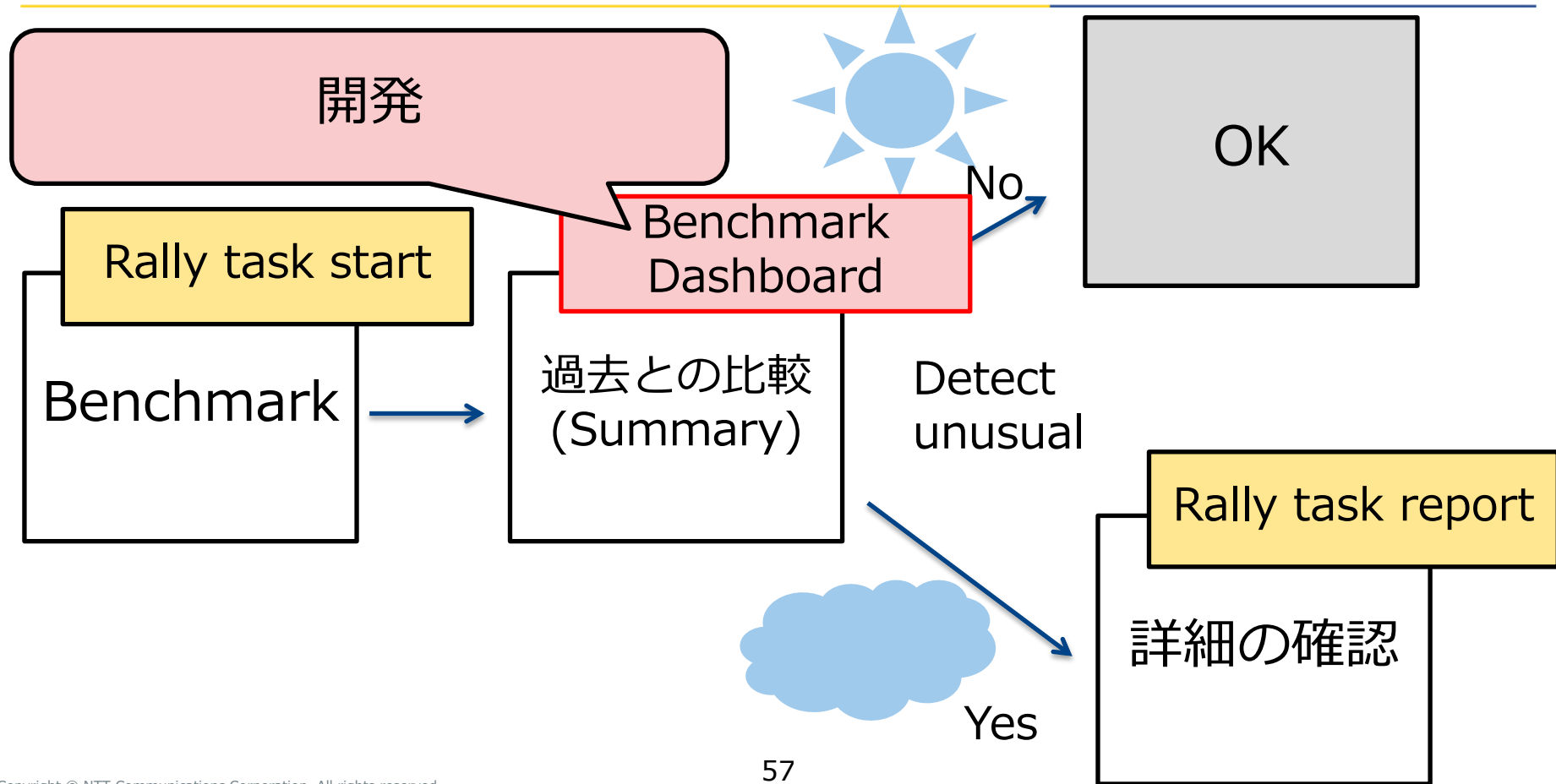
# How I use Rally

Benchmark



# How I use Rally

Benchmark



## Benchmark Dashboard

- 2つの機能を提供
  - ✓ 過去のサマリーと現在のサマリーを比較
  - ✓ 詳細レポートへのリンク

Benchmark Dashboard

Dashboard

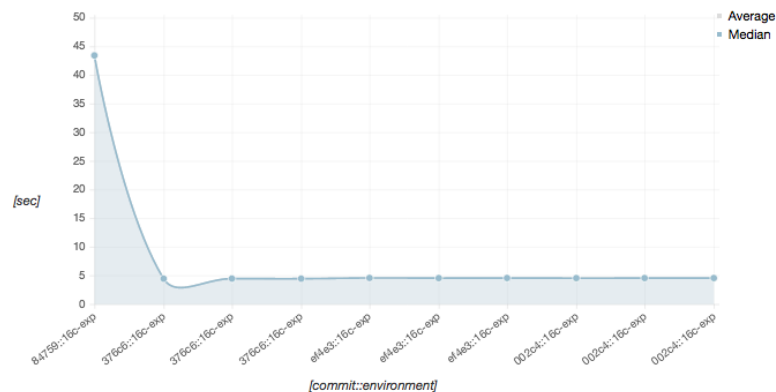
Charts

## Dashboard

boot\_server1.json

Job	commit	environment	avg	median	success	Date
225	002c421	16c-exp...	4.573	4.554	100.0%	2015/10/21 14:16
225	002c421	16c-exp...	4.561	4.549	100.0%	2015/10/21 13:45
225	002c421	16c-exp...	4.575	4.542	100.0%	2015/10/21 13:13
224	ef4e38e	16c-exp...	4.573	4.559	100.0%	2015/10/21 1:49
224	ef4e38e	16c-exp...	4.561	4.545	100.0%	2015/10/21 1:16
224	ef4e38e	16c-exp...	4.585	4.575	100.0%	2015/10/21 0:44
223	376c60a	16c-exp...	4.465	4.449	100.0%	2015/10/20 21:48
223	376c60a	16c-exp...	4.476	4.464	100.0%	2015/10/20 21:18
223	376c60a	16c-exp...	4.463	4.443	100.0%	2015/10/20 20:47
222	8475983	16c-exp...	43.473	43.416	100.0%	2015/10/20 17:38

boot\_server1.json

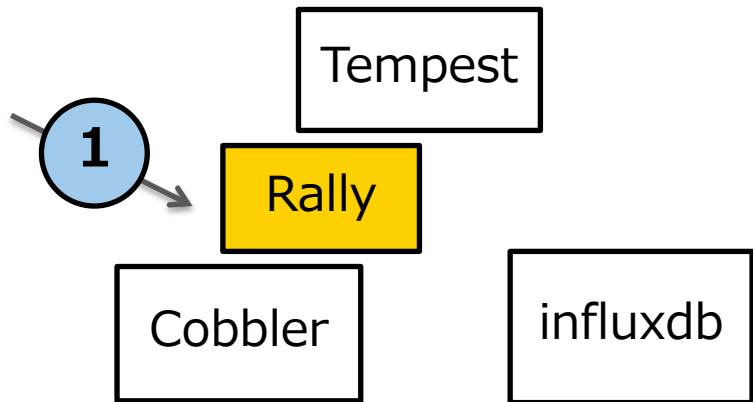
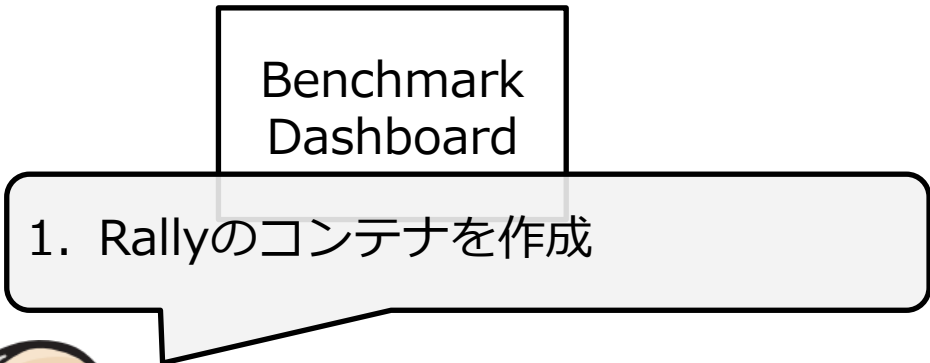
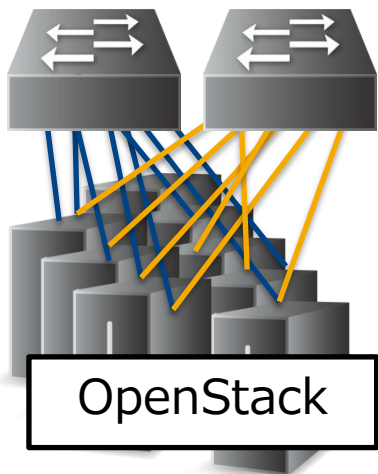
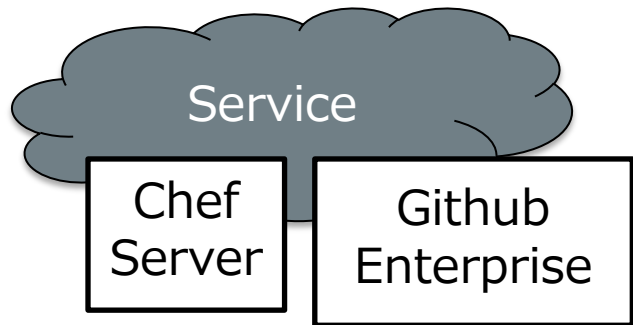


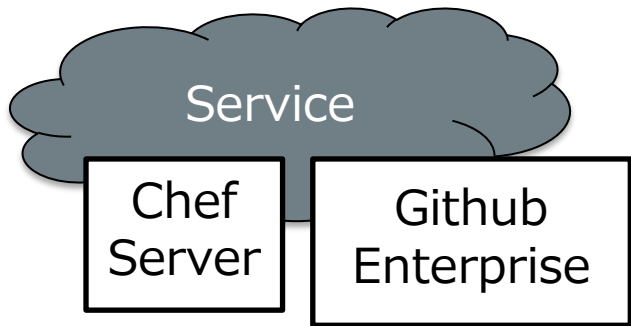
boot\_server30.json

Job	commit	environment	avg	median	success	Date
225	002c421	16c-exp...	39.334	36.14	100.0%	2015/10/21 14:16

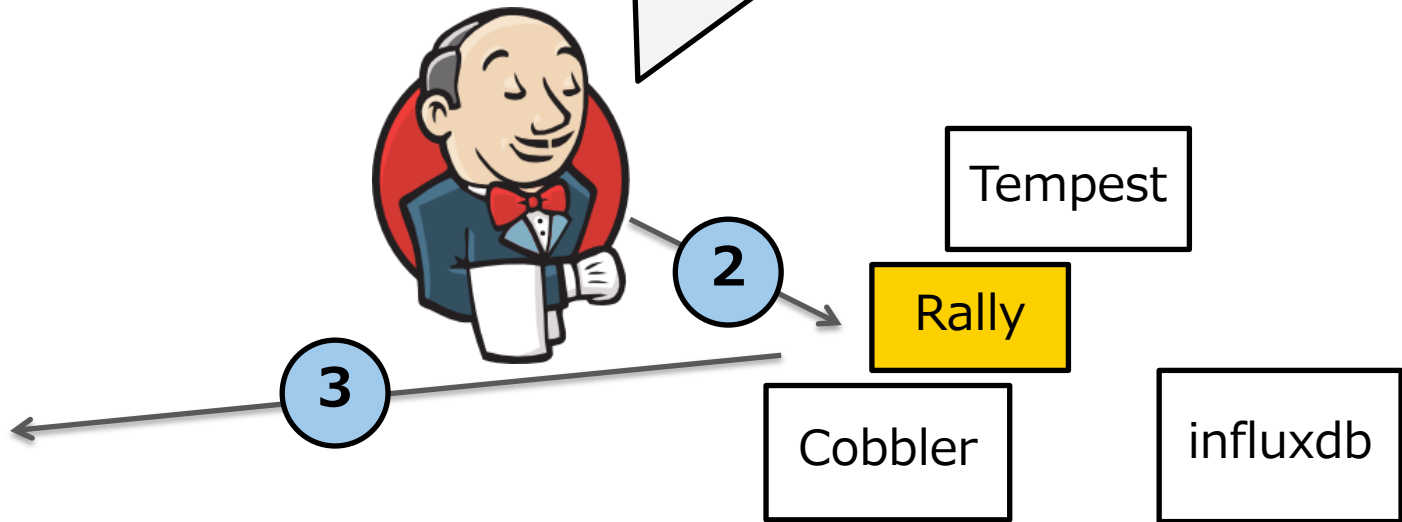
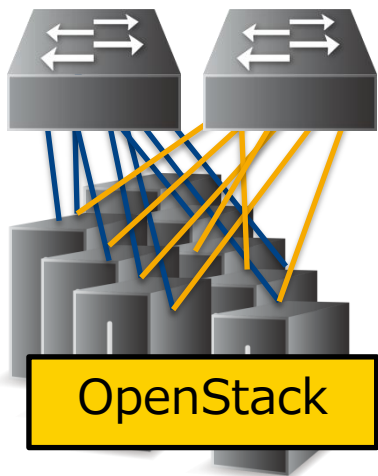
boot\_server30.json



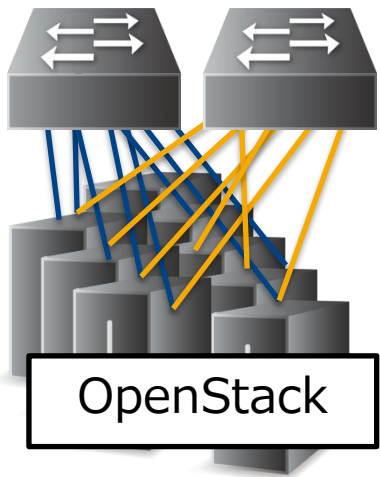




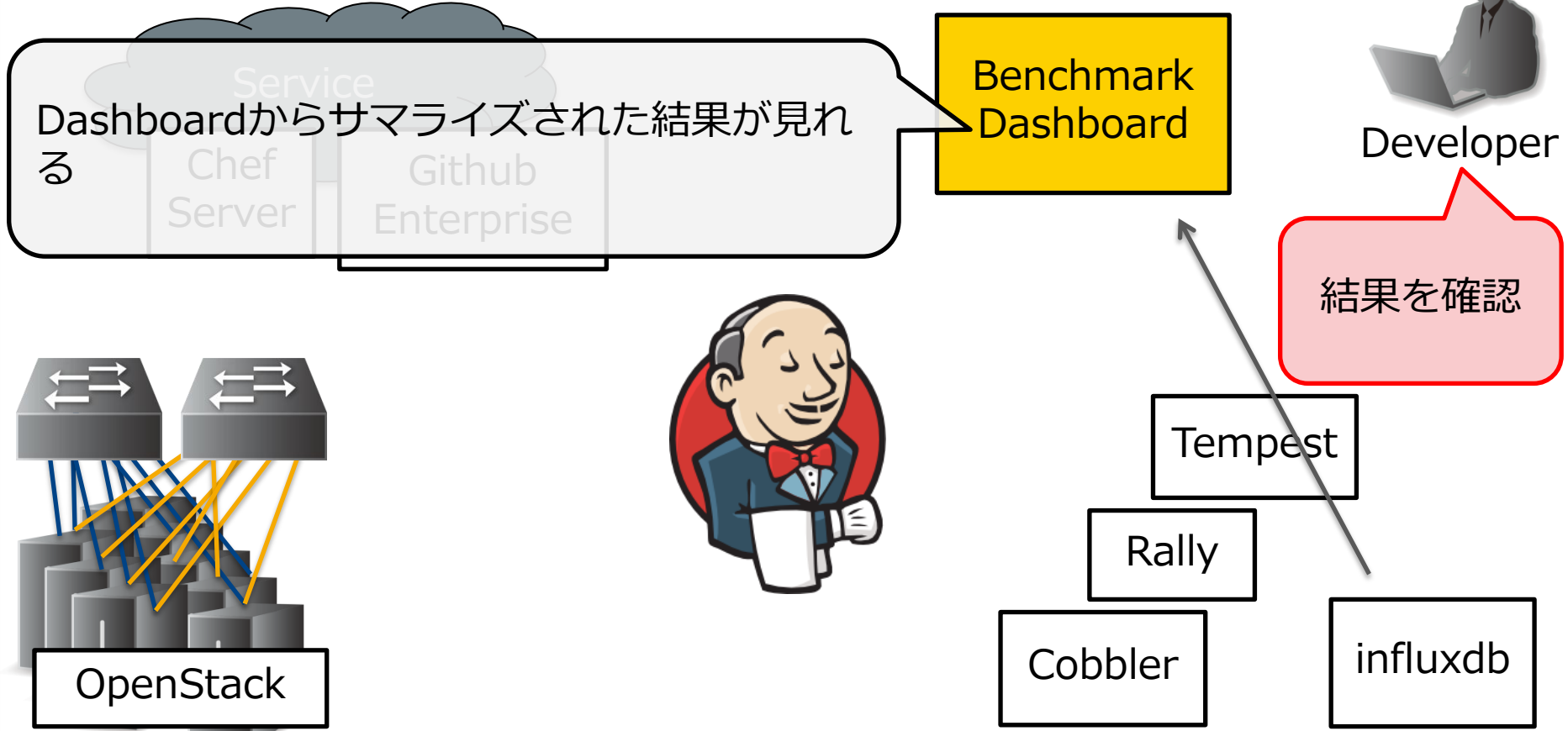
2. Register existing OpenStack  
3. ベンチマークの実行



4. report.htmlをBenchmark Dashboard  
にアップロード  
5. InfluxdbにサマリーをPOST



# Benchmark flow (4/4)



## Pros

- 過去との結果の比較ができる
- ベンチマークのプロセスがすべて自動化されている

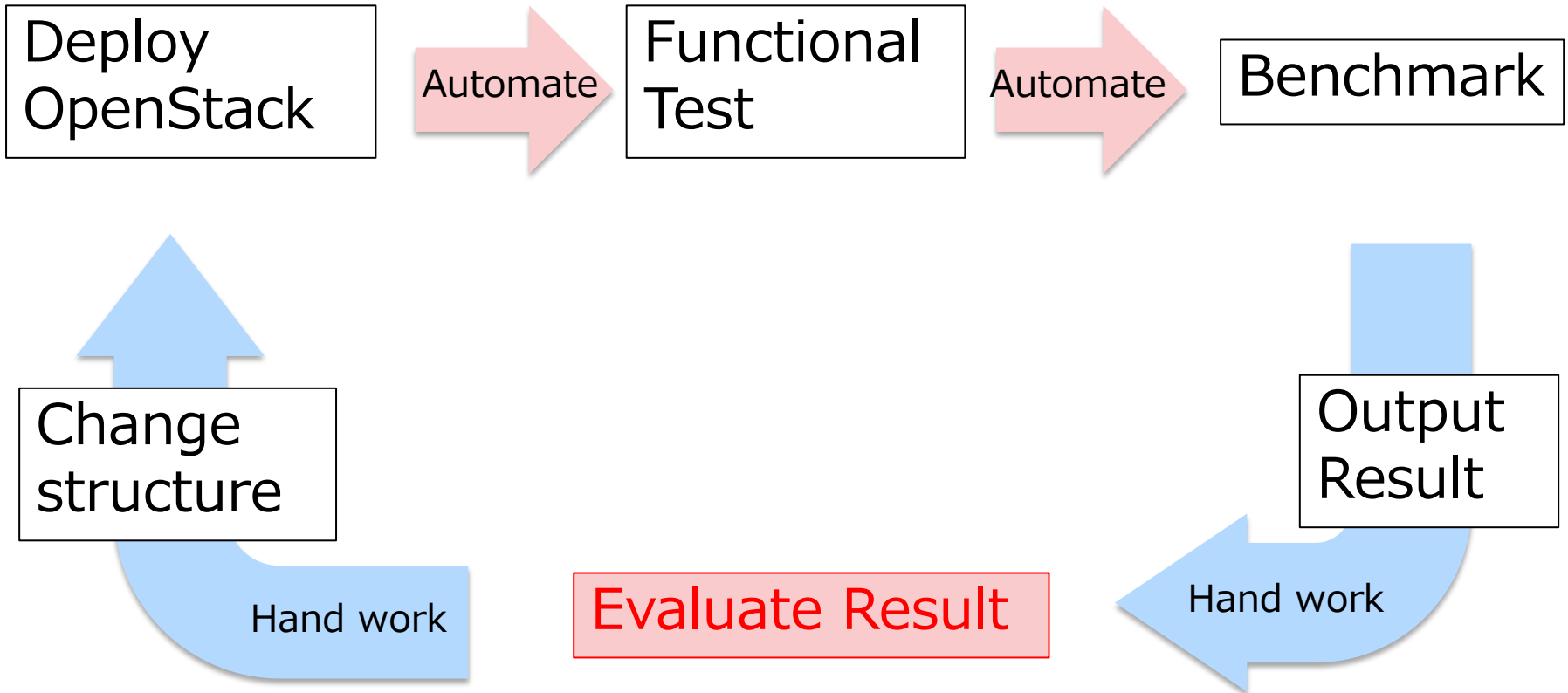
## Cons

- たくさんのシナリオはサポートしていない
- ベンチマークのベストプラクティスが明らかでない

# 結果の評価

---

# Deploy and Test and Benchmark cycle

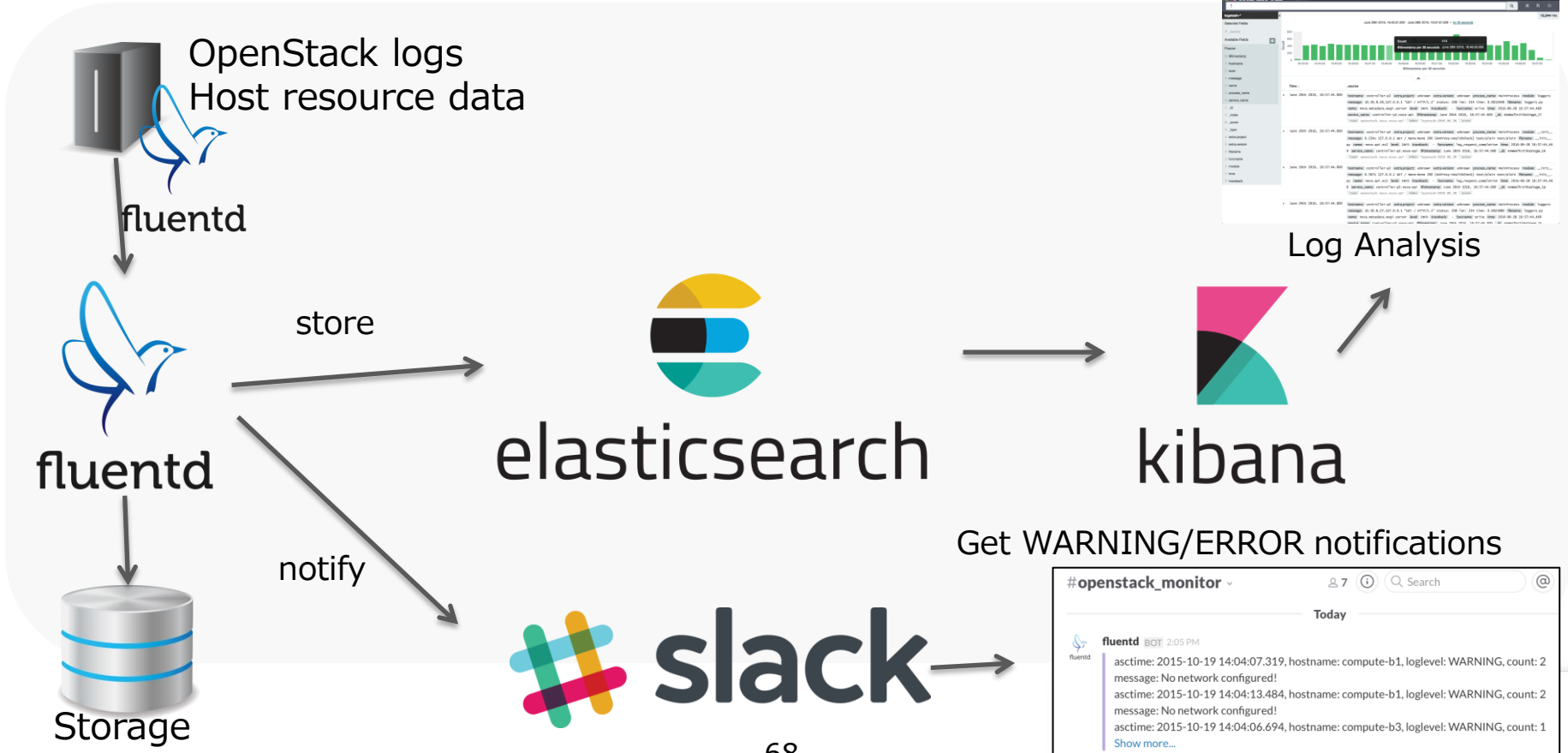


## When benchmark score is bad

---

- ボトルネックや問題をみつける必要がある
  - ✓ ホストが多すぎる
  - ✓ ログが多すぎる
  - ✓ 発見が難しい
  
- 解析基盤の構築
  - ✓ データの集約
  - ✓ モニタリング
  - ✓ アラートの通知

# Architecture to evaluate the result

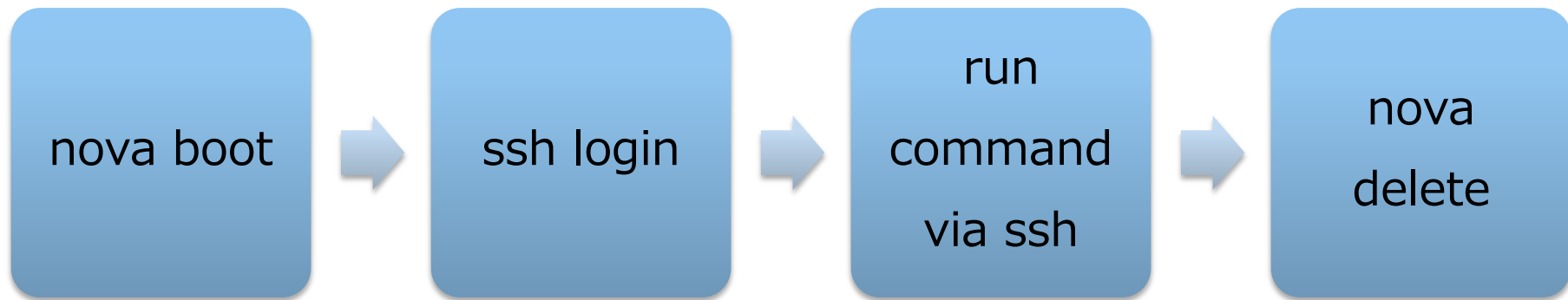


# ユースケース

---

## ベンチマークシナリオ

- VMTask scenario (Rallyにデフォルトで入っているシナリオ)
  - ✓ 違う並列数のシナリオを複数用意 (30, 50)

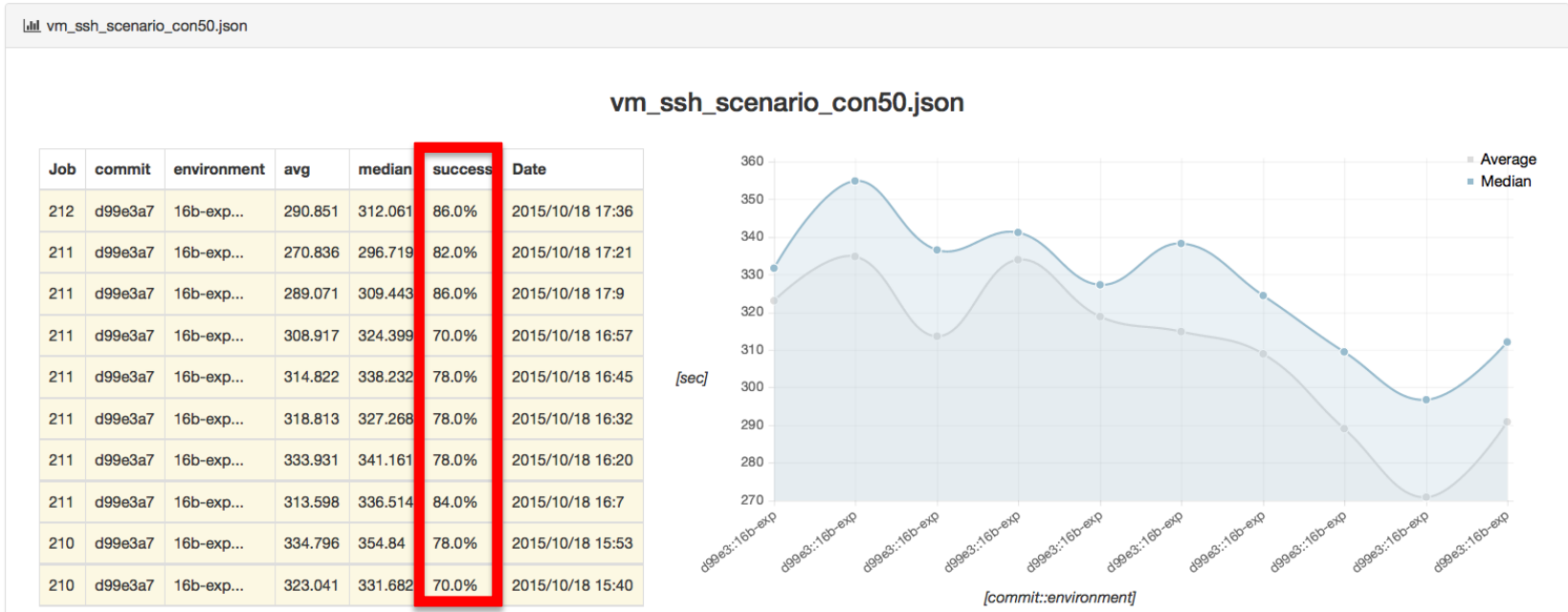


## VMTaskシナリオ

- 一般的なユースケースのシナリオ
- 総合的なシナリオ
  - ✓ インスタンス
  - ✓ セキュリティグループ
  - ✓ テナントネットワーク
  - ✓ ルーター
  - ✓ Floating IP
- ユーザーの数が増えるとどのコンポーネントに影響が出るのを知りたい

## ベンチマークで起こった問題

- 成功率が100%にならない



## ベンチマークの詳細を確認

Rally benchmark results

Benchmark overview

### VMTasks.boot\_runcommand\_delete (412.718s)

Overview Details Output Failures Input task

#### Benchmark failures (12 iterations failed)

Exception message

		Exception message
▶ 21	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.70'):<no id> to become ICMP UP current status NONE
▶ 24	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.72'):<no id> to become ICMP UP current status NONE
▶ 26	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.74'):<no id> to become ICMP UP current status NONE
▶ 31	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.76'):<no id> to become ICMP UP current status NONE
▶ 38	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.81'):<no id> to become ICMP UP current status NONE
▶ 39	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.83'):<no id> to become ICMP UP current status NONE
▶ 40	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.86'):<no id> to become ICMP UP current status NONE
▶ 45	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.95'):<no id> to become ICMP UP current status NONE
▶ 46	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.105'):<no id> to become ICMP UP current status NONE
▶ 47	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.108'):<no id> to become ICMP UP current status NONE
▶ 48	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.110'):<no id> to become ICMP UP current status NONE
▶ 49	TimeoutExcepti	Rally tired waiting for IPAddress IPAddress('10.0.4.113'):<no id> to become ICMP UP current status NONE

IPのリーチャビ  
リティがない

# Neutronのlogをチェック

^

	service_name	level	message
100	controller-p2. <b>neutron</b> -server	ERROR	TimeoutError: QueuePool limit of size 5 overflow 20 reached, connection timed out, timeout 10
100	controller-	INFO	10.50.8.3 [2016/07/15/2016 16:33:11] "GET

connection poolがオーバーフローしている。  
DBの設定について見直す必要。

# Change configuration

Use case

```
diff --git a/cookbooks/neutron/templates/default/neutron.conf.erb b/cookbooks/neutron/templates/default/neutron.conf.erb
index e70480a..ca9478d 100644
--- a/cookbooks/neutron/templates/default/neutron.conf.erb
+++ b/cookbooks/neutron/templates/default/neutron.conf.erb
@@ -600,23 +600,23 @@ connection = mysql://<%= node[:db][:user] %>:<%= node[:db][:password] %>@<%= nod

# Database reconnection retry times - in event connectivity is lost
# set to -1 implies an infinite retry count
-max_retries = 10
+max_retries = 300

# Database reconnection interval in seconds - if the initial connection to
# database fails
-retry_interval = 10
+retry_interval = 2

# Minimum number of SQL connections to keep open in a pool
-min_pool_size = 1
+min_pool_size = 10

# Maximum number of SQL connections to keep open in a pool
-max_pool_size = 5
+max_pool_size = 40

# Timeout in seconds before idle sql connections are reaped
-idle_timeout = 3600
+idle_timeout = 480

# If set, use this value for max_overflow with sqlalchemy
-max_overflow = 20
+max_overflow = 60
```

[database] in neutron.confを変更

- max\_retries
- retry\_interval
- min\_pool\_size
- max\_pool\_size
- idle\_timeout
- max\_overflow

```
$ git checkout -b change_db_setting_in_neutron_conf  
$ git commit -am "foobar"  
$ git push origin change_db_setting_in_neutron_conf
```

SlackからJenkinsに  
リクエスト



jenkins BOT 9:52 PM ★

Github to Chef-Server - #53 ユーザー-anonymousが実行 (Open)



@jenkins: reflect chefserver our-openstack branch=change\_db\_setting\_in\_neutron\_conf

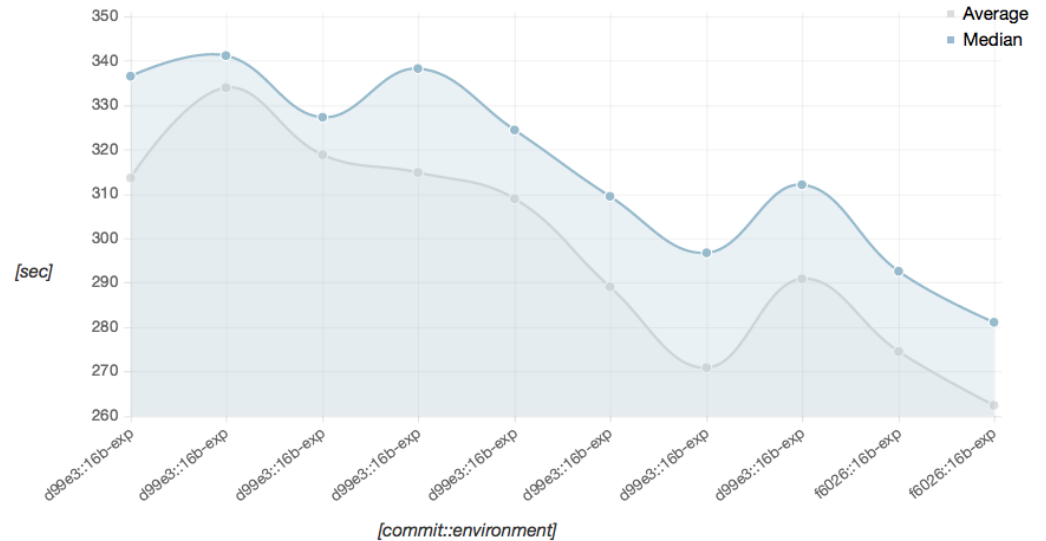


\*bold\* *italics* ~strike~ `code` ```preformatted``` >quote

設定変更によって成功率が改善

vm\_ssh\_scenario\_con50.json

Job	commit	environment	avg	median	success	Date
213	f60261e	16b-exp...	262.363	281.061	100.0%	2015/10/18 18:21
213	f60261e	16b-exp...	274.511	292.565	100.0%	2015/10/18 17:57
212	d99e3a7	16b-exp...	290.851	312.061	86.0%	2015/10/18 17:36
211	d99e3a7	16b-exp...	270.836	296.719	82.0%	2015/10/18 17:21
211	d99e3a7	16b-exp...	289.071	309.443	86.0%	2015/10/18 17:9
211	d99e3a7	16b-exp...	308.917	324.399	70.0%	2015/10/18 16:57
211	d99e3a7	16b-exp...	314.822	338.232	78.0%	2015/10/18 16:45
211	d99e3a7	16b-exp...	318.813	327.268	78.0%	2015/10/18 16:32
211	d99e3a7	16b-exp...	333.931	341.161	78.0%	2015/10/18 16:20
211	d99e3a7	16b-exp...	313.598	336.514	84.0%	2015/10/18 16:7



# まとめ

---

- デフォルトの設定はすべてのユースケースにはマッチしない
  - ✓ 最適な設定を見つける必要がある
- 最適な設定を見つけるにはサイクルを回す必要がある
  - ✓ デプロイ, テスト, ベンチマーク, 評価
- すべてのプロセスを自動化してくれるような魔法のツールはない
  - ✓ 複数のOSSを組み合わせることで実現可能

# Rally最新動向

---

- 機能のPlugin化
  - TaskやDeploymentなどの基本的な機能を含め、Plugin化
  - あわせてPluginに関するドキュメントや、helpコマンドでpluginのhelpを出せる機能
- 結果のエクスポート機構
  - サンプルとしてjsonファイルでベンチマークの結果を出力する機能
  - Pluginを書けばサードパーティのツールに送ることも
- DBマイグレーション機能

- Output画面のカスタマイズ機能
- マルチAPIバージョンサポート
  - Taskの中でAPI Versionがコントロールできる
- Heat based VM Workloads
  - Heatでシステムを構築し、そのシステム上でworkloadを流して結果をRallyに収集できるようになった  
OpenStack上に複数台から構成されるシステムを想定したベンチマークが可能に

- テストのシナリオ指定
  - Tempestのテスト名を羅列したファイルを読み込むことで実行するテストを指定できる
- Tempestの結果読み込み機能
- Xfail機構
  - 失敗することがわかっているテストを指定することでそのテストが失敗しても成功とみなす
- 外部のtempest.conf読み込み機能

- 将来的に実装予定の機能
  - Tempest Pluginのサポート
  - Rally Task Certification
    - Rally Certificatedなベンチマークシナリオリスト
  - Trend & Compare reports
    - 結果の比較機能
  - HA testing
    - 破壊的なシナリオ
  - Rally as a Service

# License

---



Jenkins image by Jenkins project is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.