

OSC2017 Chiba



Ansibleによる自動構築 ベストプラクティスをOSSで公開

配布用

2017年9月2日
TIS株式会社 OSS推進室
溝口 則行

Agenda

- 自己紹介 & 私たちのチームの紹介
- 私たちが目指していること
- **SHIFT紀元前:**
OSSミドルウェアのおすすめパターン化
- **Project SHIFT & SHIFT ware**
- **“SHIFT ware”公開**
- まとめ

溝口 則行, TIS株式会社 IT基盤技術本部 OSS推進室長

～2000年代前半:

LISPでエキスパートシステム型ビジネスアプリの開発
UNIXとC言語を中心にしたシステム開発
消費者向けインターネットサイトのシステム構築, 運用保守

2000年代前半～:

TIS戦略技術センター(現組織名)にて技術コンサルティング
得意分野: ・Linux/UNIX系システムでのミドルウェア
・性能エンジニアリング
ミドルウェアを中心にしたOSS活用推進グループのリーダー

2015年度～:

OSS推進室 室長

その他:

- ・オープンソースビジネス推進協議会(OBCI) 理事
- ・OSSコンソーシアム 理事
- ・OSS運用管理勉強会 世話人
- ・経済産業省所管 情報処理技術者試験 試験委員
- ・執筆: @IT 連載「性能エンジニアリング入門」, 他

http://www.atmarkit.co.jp/fnetwork/index/index_perform.html



私たちのチーム — OSSコミュニティへの貢献：Zabbixの情報発信、機能強化



HyCLOps for Zabbix

概要

HyCLOpsはZabbixの拡張ツールです。Amazon EC2やVMware vSphereの環境
理することを実現します。

クラウド環境を監視するための
拡張機能をOSS公開

HyCLOpsはZabbixの拡張ツールです。Amazon EC2やVMware vSphereの環境
理することを実現します。

ZABBIX-JP

Japanese Zabbix Community

ログイン アカウ

HOME NEWS FORUM DOCUMENTS DEMO CONTACT

OSS運用管理勉強会

オープンソースソフトウェアの運用管理ツ
体です

コミュニティ活動への
参加・運営協力

ザビックス Zabbix 統合監視 徹底活用

TIS株式会社
池田大輔(著)
Ikeda Daikoku

複雑化・大規模化するインフラの一元管理

マニュアルだけではわからない
活用のノウハウを書籍化

物理サーバ・仮想マシン・クラウド
混在環境を効率的に自動監視する

- SNMP、IPMIを活用した物理環境の監視
- VMware監視機能によるvSphere環境の監視
- AWSのアクセス状況、課金情報の監視
- 物理・仮想・クラウド環境間の移行、連携
- マップ機能によるシステム構成の可視化
- アクション機能による障害への自動対応
- Chef連携による環境構築・運用の自動化
- Fluentd連携によるログ集約管理

技術評論社



ラトビア共和国・リガで開催された
年次大会での講演

宣伝！

私たちのチーム — OSSコミュニティへの貢献: PostgreSQLの情報発信、機能強化

災害対策のためのストリーミングレプリケーション検証
コンソーシアムWGリーダーとして検証成果を発表

発表したのはTIS IT基盤技術本部 IT基盤技術推進課の中西聡氏(写真1)。中西氏は「DRのサービスレベルは高ければよいとは限らない。災害発生時にITシステムをどのレベルで継続するのか、事前に検討しておくことがポイント」と話す。災害は起こるか分からない。目標レベルを定めてから計画する必要がある。



各種ツールの情報収集と提供、整備などの活動を通じて、ミッションクリティカル性の高いエンタープライズ領域へのPostgreSQLの普及を推進することを目的として設立された団体です。



PostgreSQL monitoring template for Zabbix (pg_monz)

PostgreSQL監視機能をOSS公開

pg_monzの目的

pg_monzを導入することで、PostgreSQLの死活監視、リソース監視、性能監視などが行えます。PostgreSQL単体で稼働するシングル構成の状態、PostgreSQLのStreaming Replicationを使った冗長構成の状態、pgpool-IIを使った負荷分散構成の状態の監視を行うことができ、PostgreSQLを運用する様々な環境の監視を行うことができます。PostgreSQL環境の監視を行えるようにすることで、障害発生時の自動復旧処理に活用できたり、長期的な運用時の状態の変化の認知に有効に働きます。

pg_monz version 1.0からの変更点

pg_monzはversion 1.0として2013年12月に初期版をリリースしました。version 1.0では、単一のPostgreSQLサーバーを監視するだけでなく、Streaming Replicationを使った冗長構成の監視も行うことができます。



PostgreSQL 9.2の同期レプリケーションを利用する際の勘所

近未来の技術トレンドを先取り！「Tech-Sketch」出張所 第3回

小林 達 (TIS株式会社) [著] 2013/05/01 14:00

いいね! 67 +1 4 51 ツイート 42 バックナンバー ブックマーク 印刷用を表示

本連載では、TIS株式会社が提供している技術ブログ「Tech-Sketch」から「コレは!」という

有力メディアを通じて検証成果を発表

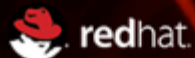


コミュニティからの表彰



宣伝!

私たちのチーム — OSSコミュニティへの貢献: Ansibleの情報発信



テクノロジー サービス & サポート 世界のお客様導入事例 Red Hat について

イベント

Ansibleではじめる自動化入門

2016年12月8日 | 東京 | レッドハット株式会社 5F セミナールーム

今年、集客力抜群な
ネタとタレント



8月
28

3社共同企画 Ansible 夏祭り

～RedHat & HPE & TIS～ in ドリコム

主催: tsarah0822

Ansible 夏祭り
～RedHat + HPE + TIS～

In ドリコム

既に満員御礼

開催前

2017/08/28(月)

18:30 ~ 21:30

Googleカレンダー icsファイル

このイベントに申し込む

開催日時が重複しているイベントに申し込んでいる場合、このイベントには申し込むことができません

募集期間

2017/07/26(水) 00:00 ~
2017/08/24(木) 21:00

イベントへの問い合わせ

宣伝!



TIS
TIS INTEC Group

×

LPI-JAPAN
Linux Professional Institute Japan



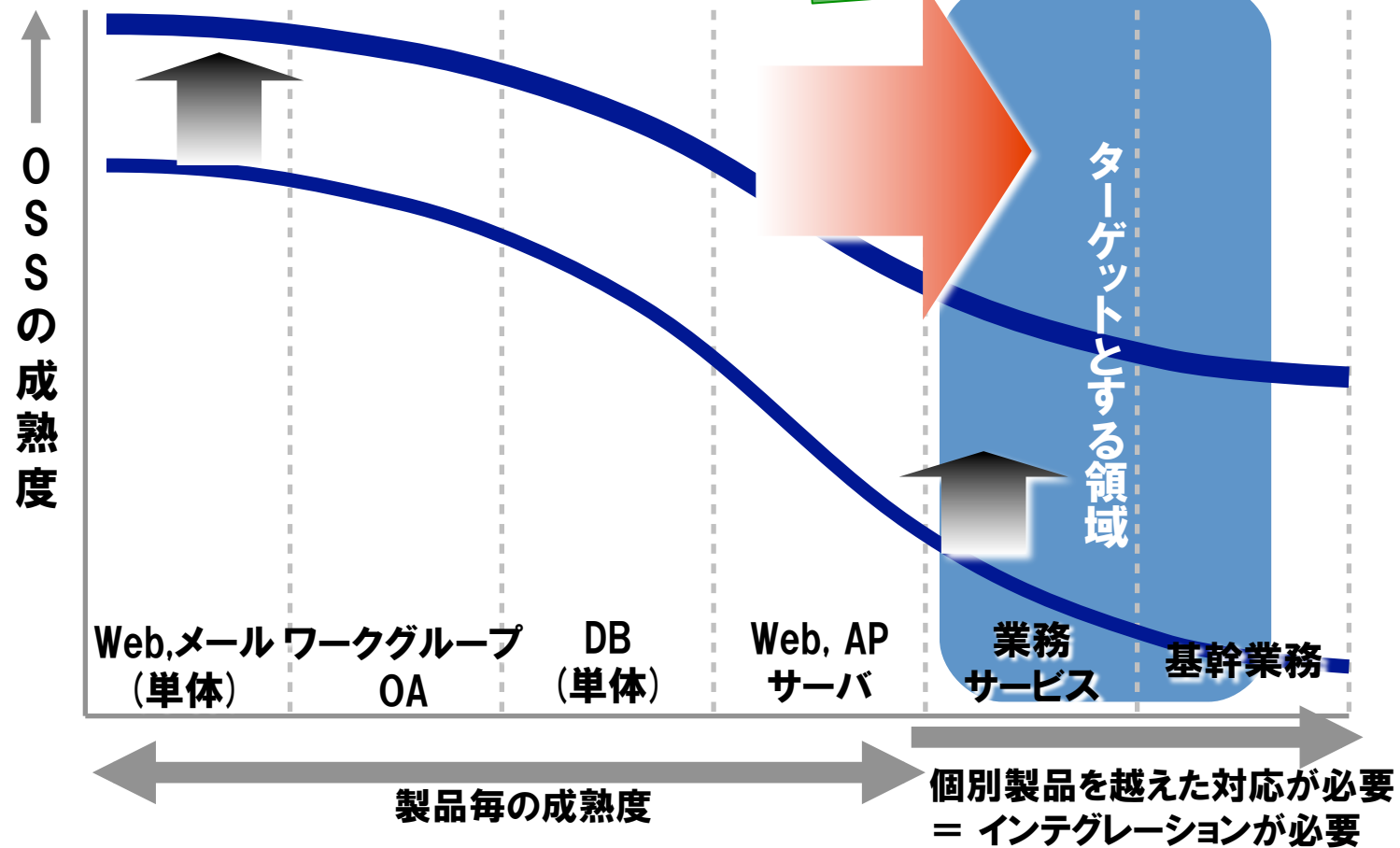
TIS & SRA OSS 共同開発



私たちが目指していること

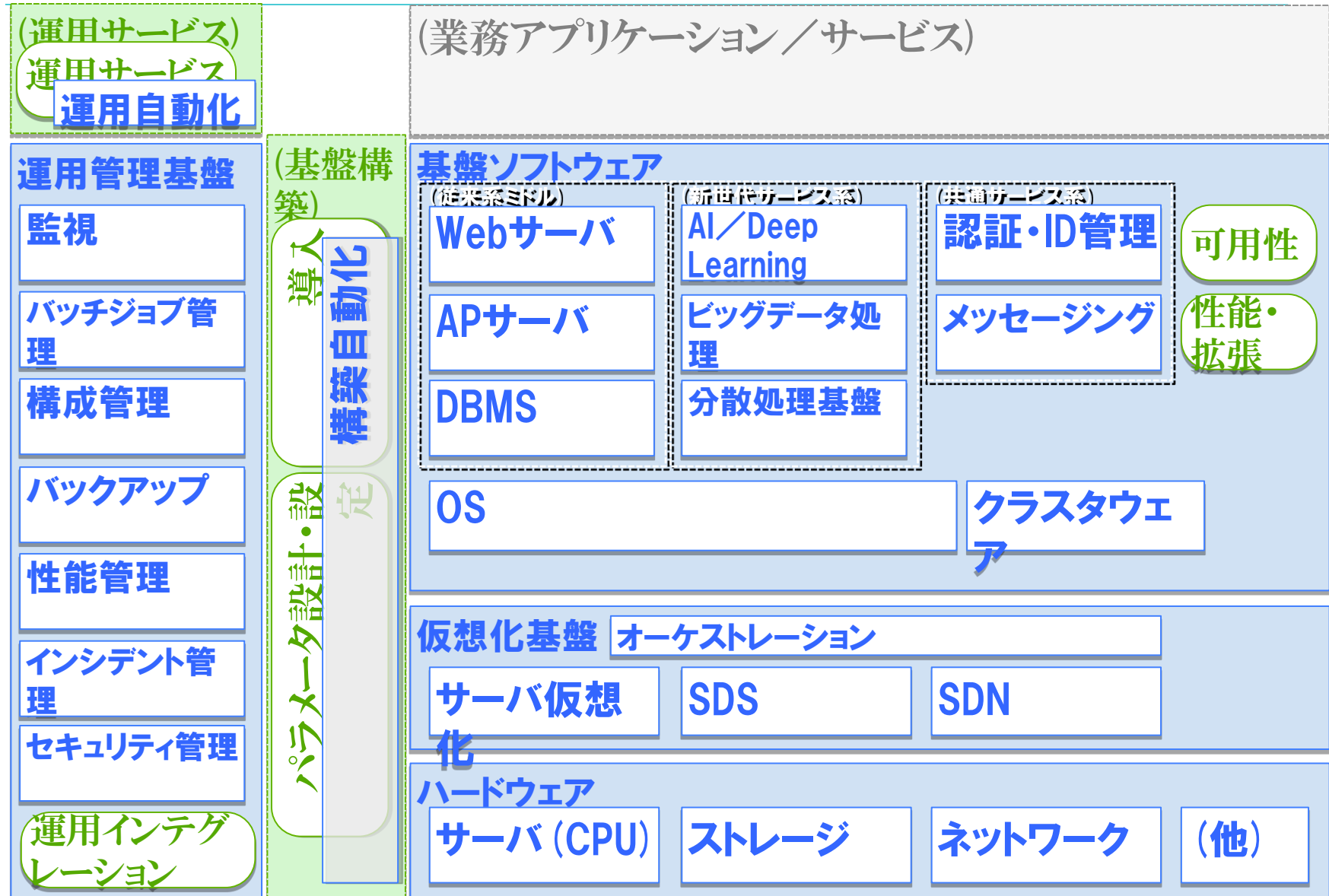
～ クラウド時代のオープンなIT基盤のために ～

OSSの活用推進の立場から…

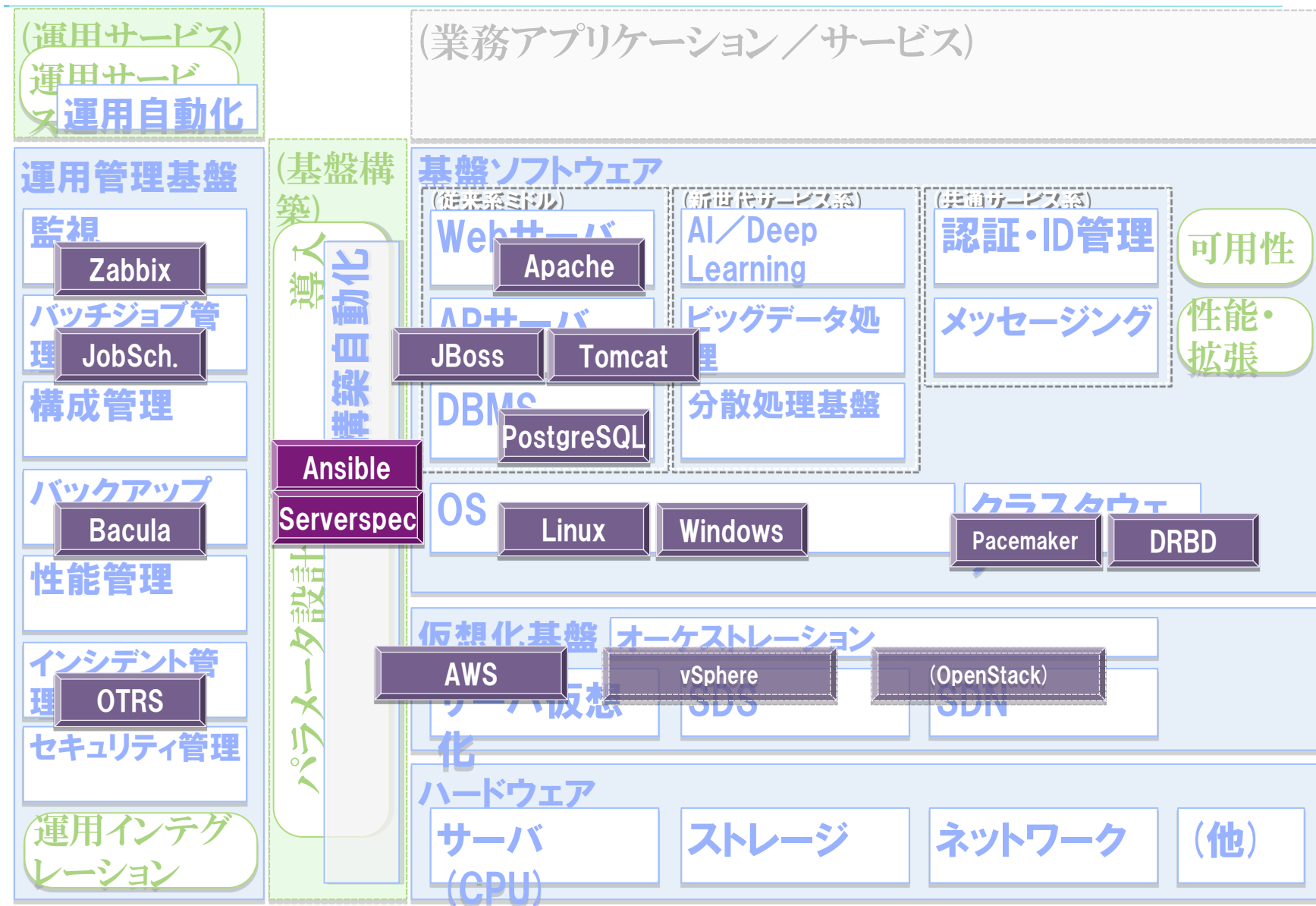


参考: IPA, 「オープンソースで構築! ITシステム導入虎の巻」
<http://www.ipa.go.jp/software/open/oss/book/toranomaki.html>

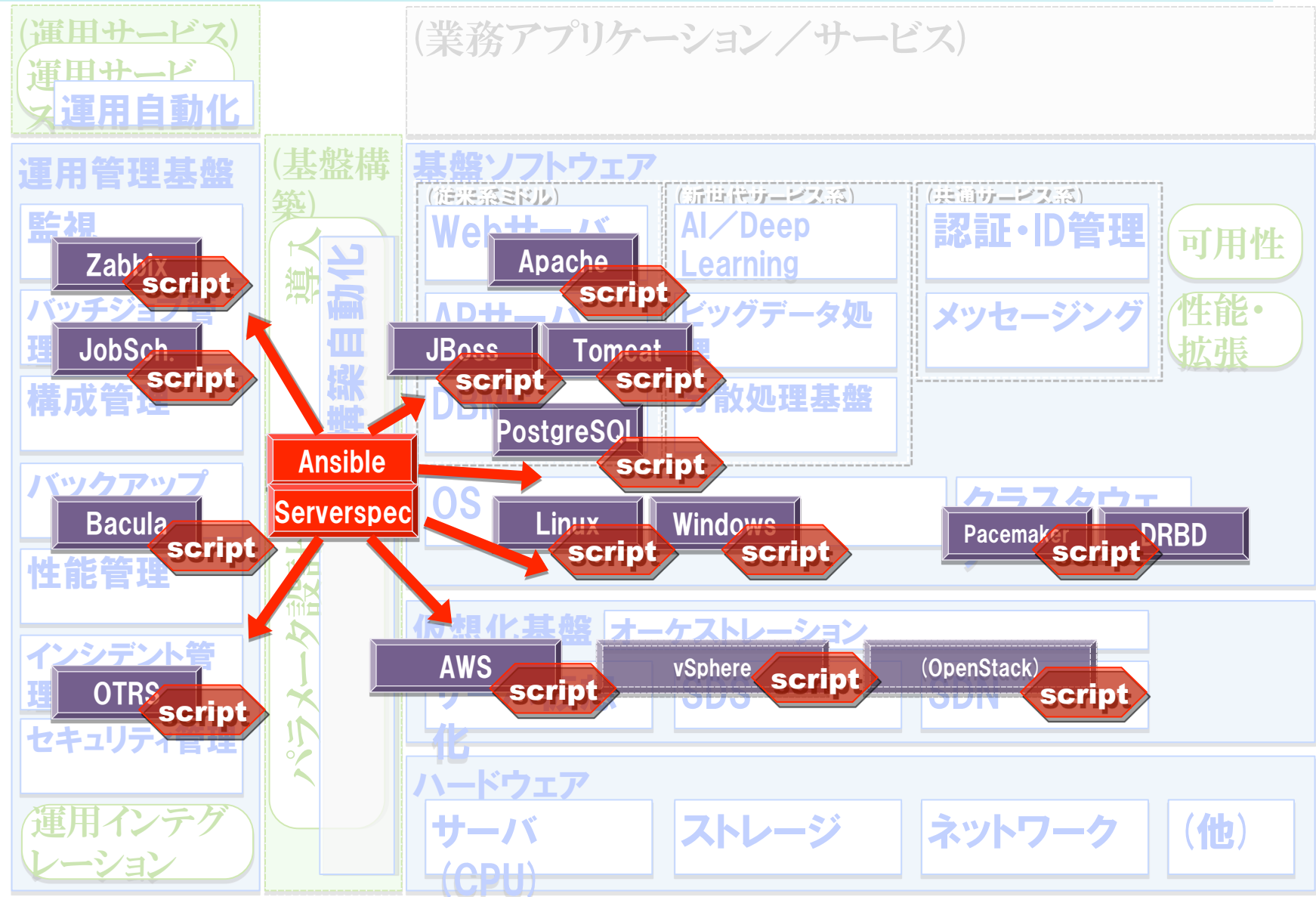
IT基盤技術参照モデル（策定中）



推奨構成パターンの一例



自動化をエンジンとして、品質と生産性の向上を目指す



SHIFT紀元前: OSSミドルウェアのおすすめパターン化 【ISHIGAKI Template】

ISHIGAKI!
TEMPLATE

2010年～

OSS推奨組合せ「ISHIGAKI Template」

このときは
“Chef”だった

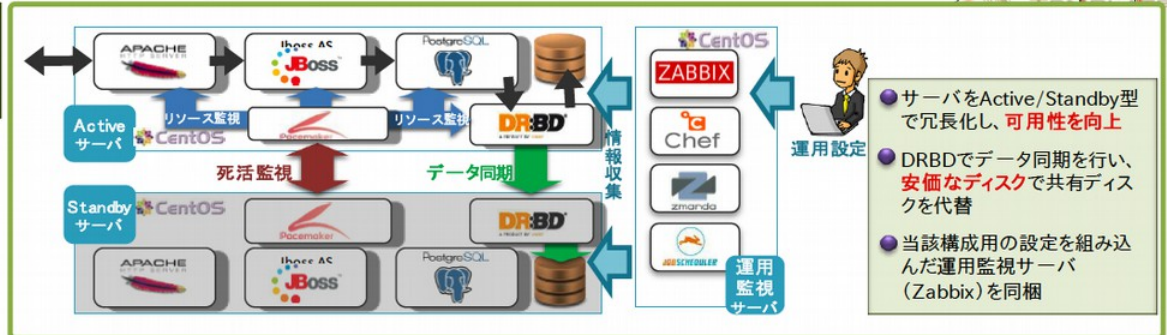


- 事前検証コストの低減**
- 導入コストの低減**
- 導入時の作業手順を説明するドキュメントを用意
→ ドキュメント不足による導入ハードル軽減
 - 選定OSSに対する推奨設定をあらかじめ適用済み【170箇所以上を設定済み】
- 運用コストの低減**
- 運用時の作業手順を説明するドキュメントを用意、設定変更、サーバの状態確認、障害時の復旧など

ISHIGAKI Template の構成パターン

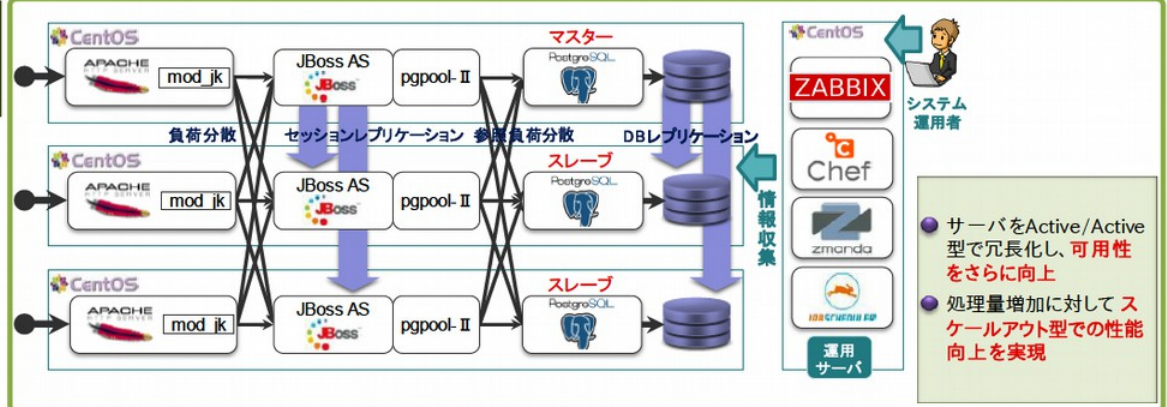
HA Edition (High-Availability)

本番運用サーバの障害時に長時間の業務停止を回避できるようHA(High-Availability)クラスタ構成を採用し、運用監視を行えるよう設計した構成です。サーバ一台で処理できる程度の比較的業務処理量の少ないケースへの適用を想定します。



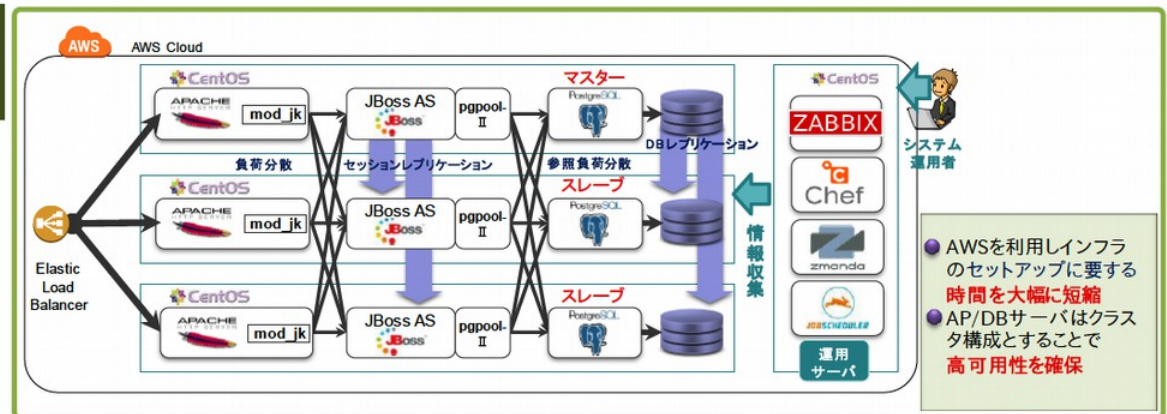
スケールアウトする Cluster Edition

一台のサーバでは処理しきれないような大量業務処理を想定した構成です。Webサーバ/アプリケーションサーバ/DBサーバそれぞれが複数台で分散して処理を行う高性能クラスタ構成をとっています。障害時の業務継続性にも対応します。



クラウドで稼働する AWS Edition

Amazon Web Service上での稼働を想定してISHIGAKI Clusterをカスタマイズした構成です。サーバ、ネットワークといったインフラにAWSの資源を利用することで、オンプレミス環境よりも素早く利用することが可能です。



Project SHIFT **& SHIFT ware**

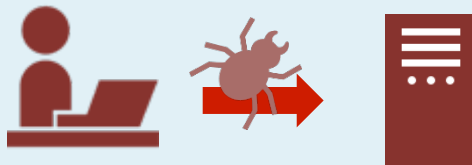
これまでのITインフラ構築・保守・運用

品質低下

属人化・引き継ぎの課題



人的ミス

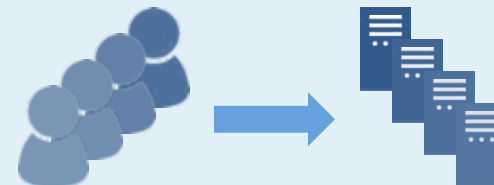


コストアップ

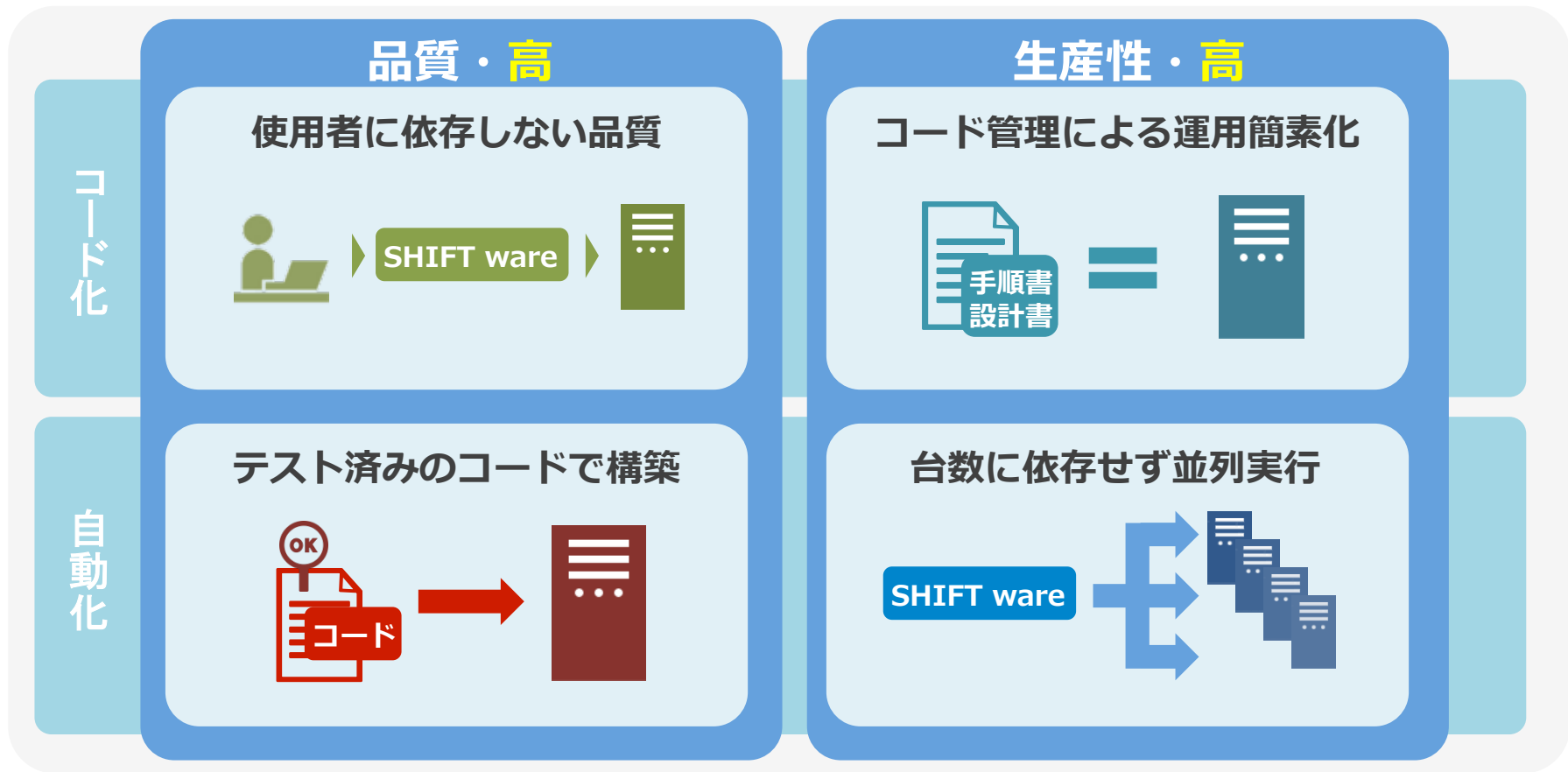
ドキュメントと実機の乖離



リニアなコスト増加



これからのITインフラ構築・保守・運用



ITインフラエンジニアの課題

だけど…

ITインフラエンジニアは、従来の課題をクリアしつつ、技術の変化にも対応しないといけない板挟み状態。

OLD



NEW



Infrastructure as Code

ITインフラの構成を、プログラムのコードで管理する手法

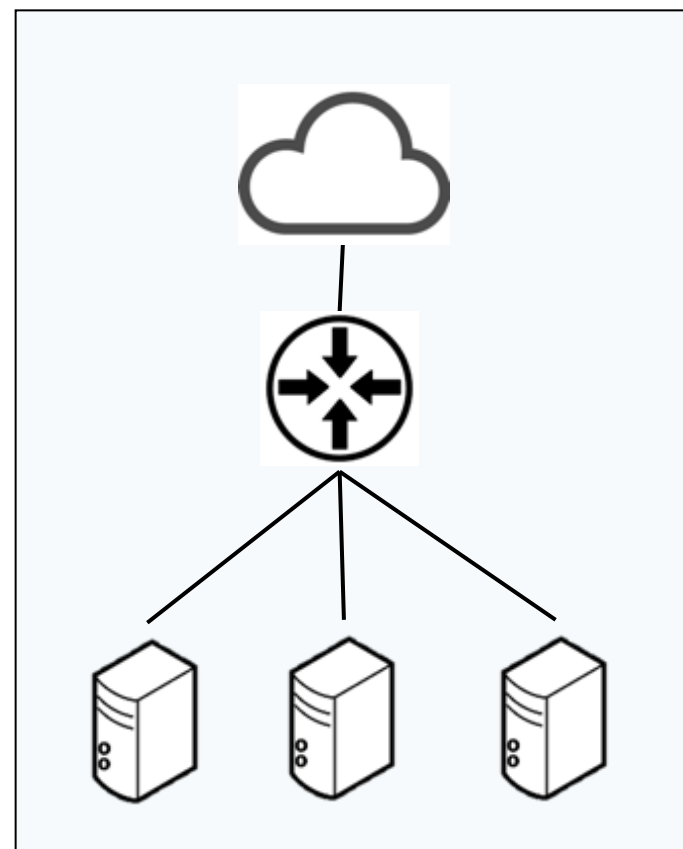
```
- name: network configuration
  os_network:
    cloud: demo
    name: demo_network

- name: subnet configuration
  os_subnet:
    cloud: demo
    name: demo_subnet
    network_name: demo_network
    cidr: 192.168.151.0/24

- name: router configuration
  os_router:
    cloud: demo
    name: demo_router
    network: provider
    interfaces: demo_subnet

- name: instance configuration
  os_server:
    cloud: demo
    name: demo_instance_{{ item }}
    image: cirros
    flavor: m1.nano
    network: demo_network
    with_items: [1,2,3]
```

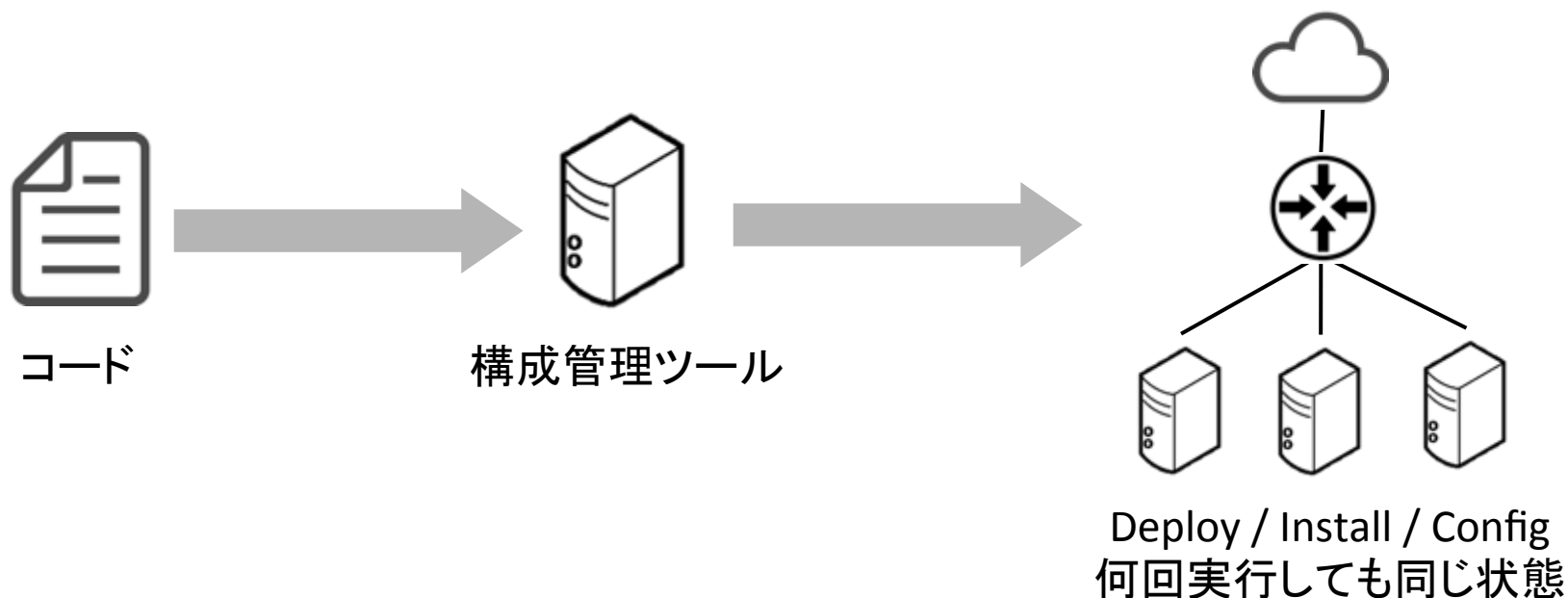
=
イコール



Infrastructure as Code

構成管理ツールによって、コードに基づきインフラが構成される。

コードは何度実行しても同じ状態に収束するため(冪等性)、コード＝構成である。



laC推進に向けての課題

しかしながら、すぐにlaCを実践できるわけではない。



やり方が変わる
コードアレルギー
学習・育成コスト
時間がない ということい訳



TISの取り組み

課題に対する2方向のアプローチ。たぶんこれが王道。

ハードルDOWN ↓

コードライブラリ
フレームワーク
スクリプト
ツール

スキルUP ↑

社内勉強会
セルフハンズオン
ナレッジ公開
案件利用支援

SHIFT ware とは

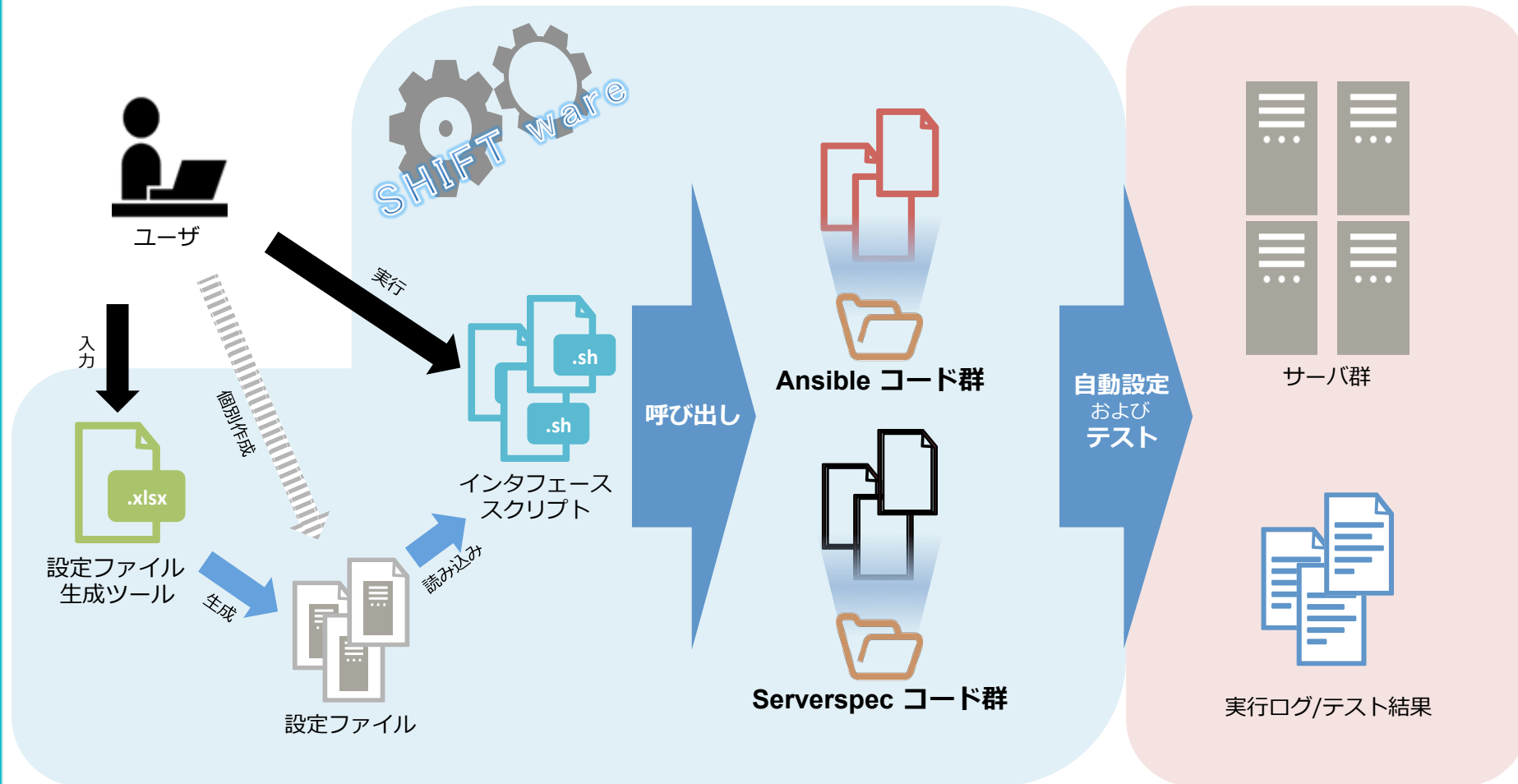
Infrastructure as Codeを誰でも利用しやすく



ハードルDOWN ↓
コードライブラリ
フレームワーク
スクリプト
ツール

スキルUP ↑
社内勉強会
セルフハンズオン
ナレッジ公開
案件利用支援

SHIFT ware とは(全体像)



SHIFT ware とは

実体は

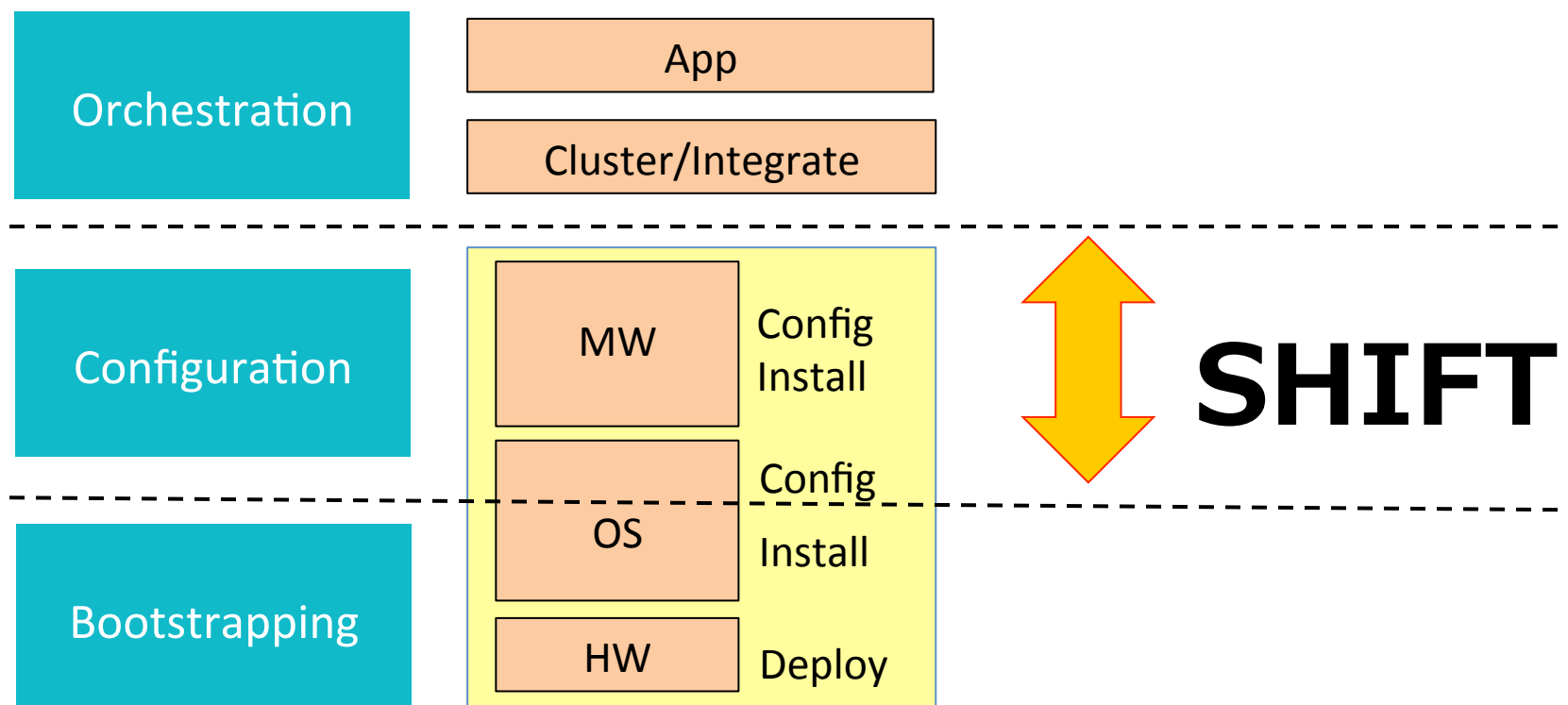
Ansible、Serverspec のコード集

コードライブラリを利用するための
フロントエンドツール



SHIFT ware で出来ること

OSの、
 自動設定 / 自動テスト
 ミドルウェアの、
 自動インストール / 自動設定 / 自動テスト



提供するもの - Linux Roles

Linux Roles

- 1-0001_Base
- 1-0102_apache
- 1-0103_Tomcat
- 1-0104_postgreSQL
- 1-0105_openJDK
- 1-0106_PrivateCA
- 1-0201_RedHatSatellite
- 1-0401_JP1Agents
- 1-0501_OracleClient
- 1-0502_OracleJRE
- 1-0701_LogstorageAgent
- 1-0801_DynatraceAgent
- 1-0901_DeepSecurityAgent
- 1-1001_UnixAgentForLDAPManager
- 1-1101_ES1Acquire
- 1-1102_ES1Logscn

Linux Base Tasks

1-0001-04_Hosts	1-0001-27_Kdump
1-0001-05_Resolve	1-0001-28_Keyboard
1-0001-07_Service	1-0001-29_Timezone
1-0001-08_Ntp	1-0001-40_Grub
1-0001-10_Selinux	1-0001-02_Interface
1-0001-12_Crontab	1-0001-03_Network
1-0001-14_Lang	1-0001-22_Route
1-0001-17_Logrotate	1-0001-39_Bonding
1-0001-20_Nsswitch	1-0001-35_SetDirectory
1-0001-23_Sshd	
1-0001-39_Snmp	
1-0001-40_Syslog	
1-0001-13_OSGroup	
1-0001-16_PasswordRules	
1-0001-37_SystemRunLevel	
1-0001-38_InterfaceNmcli	
1-0001-24_OSUser.yml	

※公開するものとは差異があるかもしれません

提供するもの - Windows Roles

Windows Roles

2-0001_Base
2-0101_Cygwin
2-0401_JP1Agents
2-0501_OracleClient
2-0701_LogstorageAgent
2-0901_DeepSecurityAgent
2-1002_LDAPManagerWindowsAgent
2-1101_ES1Acquire
2-1102_ES1Logscn

Windows Base Tasks

2-0001-002_Memory	2-0001-095_Feature
2-0001-011_User	2-0001-096_Service
2-0001-012_UserGroup	2-0001-098_Rdp
2-0001-013_Uac	2-0001-102_ErrorReport
2-0001-024_Timezone	2-0001-103_WinUpdate
2-0001-025_RecoverOs	2-0001-104_Owner
2-0001-051_Directory	2-0001-105_EventLog
2-0001-071_Hostname	2-0001-106_Registry
2-0001-072_Interface	2-0001-107_PsExecPolicy
2-0001-073_Routing	2-0001-109_Organization
2-0001-074_Firewall	
2-0001-075_Teaming	
2-0001-076_Domain	
2-0001-077_IPv6Disable	
2-0001-078_DnsSuffix	
2-0001-079_NameResolve	

※公開するものとは差異があるかもしれません

提供するもの - Excel2YAML

SlerはExcelが大好きらしい

「**Excel2YAML**」とは、Ansible、ServerspecのパラメータファイルをExcelで作成できるツール

(おまけとして作ったが、ウケがよく、需要は高い)

提供するもの - Excel2YAML

Property…設定・テストのパラメータを定義するファイル



No.	設定項目	設定値(Ansible)	確認値(ServerSpec)
1	接続用) ホスト名/IPアドレス	192.168.127.10	192.168.127.10
9	ユーザ/グループ		
10	ユーザ設定		
11	↑ ユーザ名	test1	test1
12	↑ パスワード	password	
13	↑ ユーザID(uid)	501	501
14	↑ 所属グループ	test1	test1
15	↑ ホーム	/home/test1	/home/test1
16	↑ ログインシェル	/bin/bash	/bin/bash
17	↓ セカンダリグループ	test2	test2
18	↑ ユーザ名	test2	test2
19	↑ パスワード	password	
20	↑ ユーザID(uid)	502	502
21	↑ 所属グループ	test2	test2
22	↑ ホーム	/home/test2	/home/test2
23	↑ ログインシェル	/bin/bash	/bin/bash
24	↓ セカンダリグループ	test1	test1
25	グループ設定		
26	↑ グループ名	test1	test1
27	↓ グループID(gid)	503	503
28	↑ グループ名	test2	test2
29	↓ グループID(gid)	504	504
30	パスワードポリシー		
31	max_days	99999	99999
32	min_days	0	0
33	min_length	5	5
34	warn_age	7	7



```

connection_hostname: '192.168.127.10'
note_tmp_dir: '~root/.shift'
BASE:
  ID:
    user:
      - name: 'test1'
        password: 'password'
        uid: 501
        group: 'test1'
        home_dir: '/home/test1'
        shell: '/bin/bash'
        sub_groups: 'test2'
      - name: 'test2'
        password: 'password'
        uid: 502
        group: 'test2'
        home_dir: '/home/test2'
        shell: '/bin/bash'
        sub_groups: 'test1'
    user_group:
      - name: 'test1'
        gid: 503

```

こっちの方がラクなんじゃないかと思うが…

提供するもの - Excel2YAML

Inventory…実行対象（ターゲット）と実行する処理（ルール）を定義するファイル



No.	サーバネーム (orIP)	リモート ユーザ	パスワード	Ansible	Sen	Ans	Sen	Ans	Sen
1	192.168.127.10	user1	password	○	○	×	×	○	○
2	192.168.127.20	user2	password	○	○	×	×	○	○
3	192.168.127.30	user3	password	○	○	○	○	×	×
4	192.168.127.40	user4	password	○	○	○	○	×	×
5	192.168.127.50	user5	password	○	○	×	×	○	○
6	192.168.127.60	user6	password	○	○	○	○	×	×
7	192.168.127.70	user7	password	○	○	○	○	×	×
8	192.168.127.80	user8	password	○	○	×	×	○	○
9									
10									

Inventory

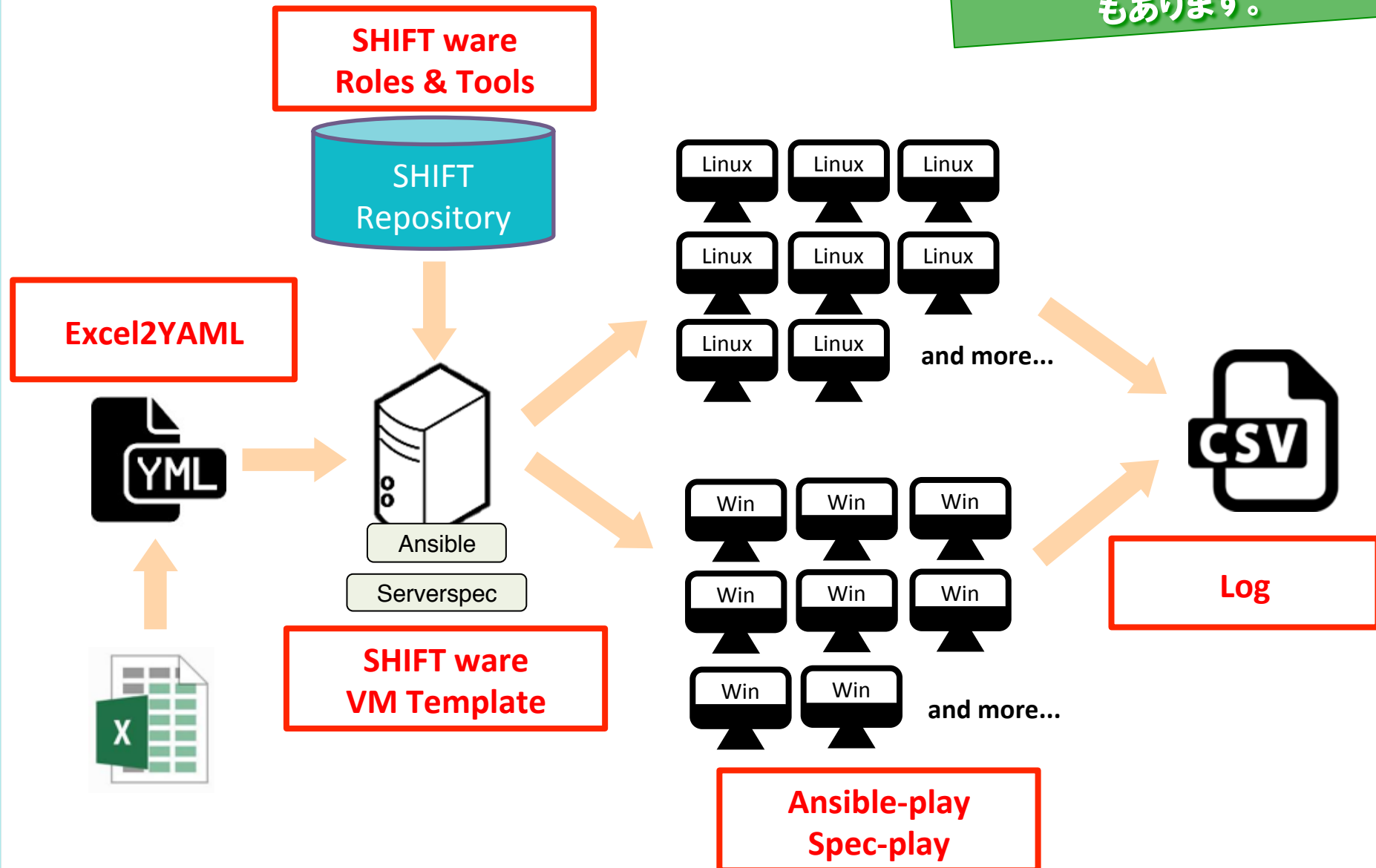
```
[1-001_Base]
192.168.127.10 ansible_user=user1 ansible_ssh_pass=password
192.168.127.20 ansible_user=user2 ansible_ssh_pass=password
192.168.127.30 ansible_user=user3 ansible_ssh_pass=password
192.168.127.40 ansible_user=user4 ansible_ssh_pass=password
192.168.127.50 ansible_user=user5 ansible_ssh_pass=password
192.168.127.60 ansible_user=user6 ansible_ssh_pass=password
192.168.127.70 ansible_user=user7 ansible_ssh_pass=password
192.168.127.80 ansible_user=user8 ansible_ssh_pass=password
[1-0102_apache_GROUP]
[1-0103_Tomcat_GROUP]
[1-0104_postgreSQL_GROUP]
[1-0105_openJDK_GROUP]
[1-0106_PrivateCA_GROUP]
192.168.127.30 ansible_user=user3 ansible_ssh_pass=password
192.168.127.40 ansible_user=user4 ansible_ssh_pass=password
192.168.127.60 ansible_user=user6 ansible_ssh_pass=password
192.168.127.70 ansible_user=user7 ansible_ssh_pass=password
[1-0201_RedHatSatellite_GROUP]
192.168.127.10 ansible_user=user1 ansible_ssh_pass=password
192.168.127.20 ansible_user=user2 ansible_ssh_pass=password
192.168.127.50 ansible_user=user5 ansible_ssh_pass=password
192.168.127.80 ansible_user=user8 ansible_ssh_pass=password
```

※AnsibleはYAMLではない

これは…
ちょっとわかる！

SHIFT ware を使ってカンタンlaC

今回は説明を省略したもの
もあります。



“SHIFT ware”公開

公開のねらい

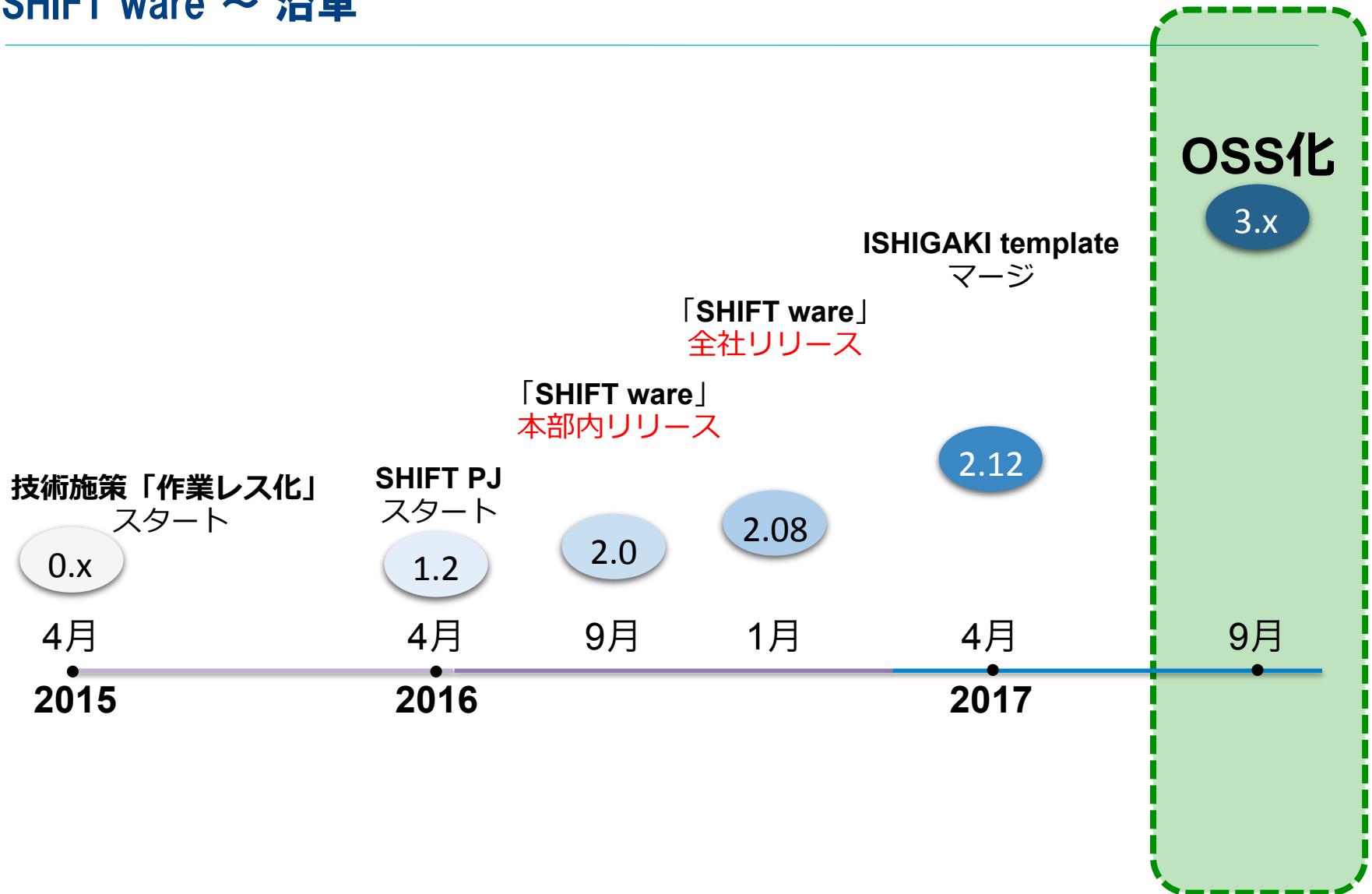
- **Infrastructure as CodeにおけるSler(ならでは)としてのSI市場でパイオニア地位確立**
TISのSlerとして、他社にない強みはなんだ？と言われた時に、より具体的に表現できるものの1つとして確立したい。

市場価値向上

- **労働生産性に関連する国策への社会貢献 ～ITエンジニアの働き方改革への貢献～**
IT企業(日本のSlerは特に)の長時間労働が問題視されているのは明らかで、それに対して、元々品質生産性向上の施策として活動していた本取り組みの成果物エッセンスを対外的に公開することで、同業他社やユーザ等にも影響を及ぼし、ひいては結果としてIT業界/SI市場での長時間労働の軽減や労働生産性の向上に貢献する。

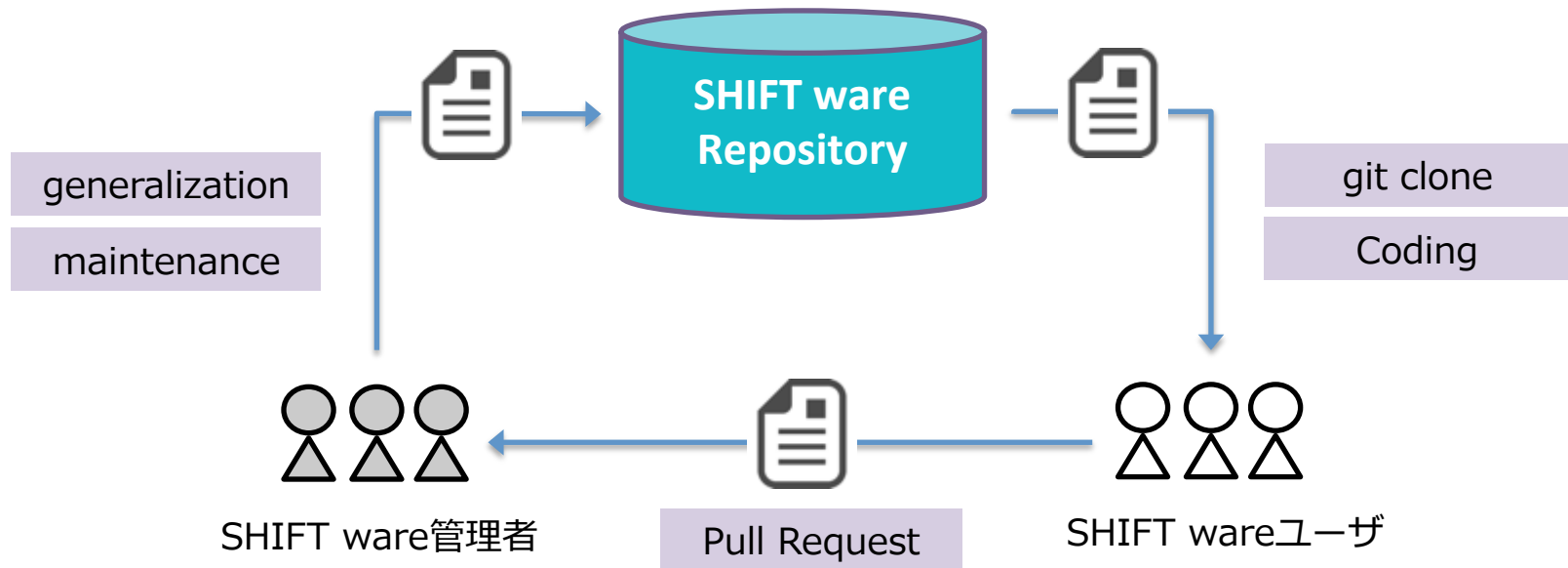
社会貢献

SHIFT ware ～ 沿革



laCのスキルアップに向けて

- ゆくゆくは、SHIFT wareユーザの誰もが
- 「Operator」から「Contributor / Committer」へ



成果の公開に向けたあれやこれや ～ 面倒なこといろいろ ～

- 「何をどこまで公開する？ 公開できる？」問題
- 承認問題
 - そもそも承認方法や承認者が決まってる？
- クオリティ問題
 - 既存コードの品質, コード規約, CIの仕組み, etc
- ライセンスは何を選ぶ？ 問題
- リポジトリをどこに置く問題（社外はGitHub！社内はどうする？）
- 名称と商標
- コミュニティの形成（立ち上げ）
- プロモーション
- 公開しちゃった後どうする？ 問題

世間に出すならクオリティもそれなりにしたい

おまけ

★ “SHIFT ware” 公開！

2017年9月下旬(予定)

★ コミュニティ

SHIFT wareユーザーグループ

https://github.com/SHIFT-ware/shift_ware/wiki/SHIFT-ware-User-Group

すみませんが、
もうちょっとお待ちください。

★ お問い合わせなど

<https://groups.google.com/forum/#!forum/shiftware>

THANK YOU



TIS

TIS INTEC Group

| Go Beyond