Open Source 協議会 IBM i

オープンソース協議会 IBM i 秋セミナー　- Python 分科会 -

Ver.　1.0　|　2020-10-22

IBM i で Python やってみた. 1

ティアンドトラスト株式会社 北原 征夫

T&Trust Tenderness and Trust
tat.co.jp

アジェンダ

- python 概説
- インストール
- データベース・アクセス
- itoolkit
- Web サーバー連携
- PHP と比較

Pytho 概説

Python 概説

- 歴史
  - 1991年 Guido van Rossum (グイド＝ヴァンロッサム:オランダ)　v0.9 公開
  - 1994年 Python 1.0 リリース
  - 2000年 Python 2.0 リリース
  - 2008年 Python 3.0 リリース 最新の安定版 3.9 (2020/10)

Java ： 1995年公開
PHP ： 1995年公開
RPG ： 1959年開発

- ライセンス形態
  - PSFL：Python Software Foundation License
    - GPL互換
    - 改変したプログラムの配布時に改変部分の公開は不要
    - PSFL 以外のライセンスの配布物と一緒に配布可能

Python 概説

特徴

- インデント
  - 波括弧{} の代わりに コロン ： とインデント でブロックを表現
  - 行の終端にセミコロン ； は不要（改行で良い）
- インタープリター、マルチプラットフォーム
  - コンパイル不要
- pypi ：Python Package Index
  - 豊富なモジュール群を利用できる：268,148 project
  - Pip (Python Package Installer) によりインストール
- PEP：Python Enhancement Proposales
  - Python の決め事や標準化などが記されている
    - PEP8：Python コーディング規約

Python

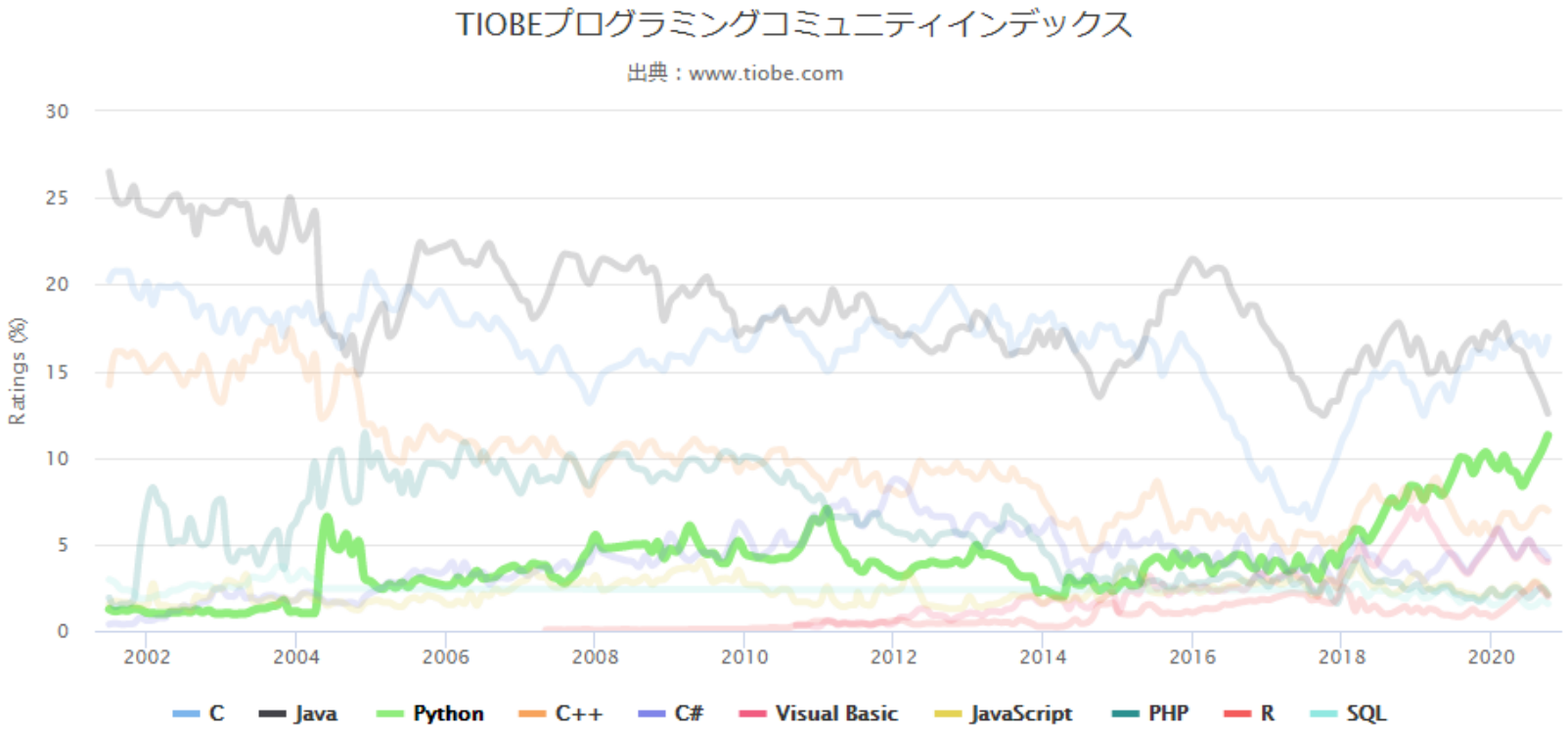
```python
if night:
    print("Hello Night World")
else:
    print("hello Morning Wrold")
```

PHP

```php
if(night){
    print("Hello Night World");
} else {
    print("Hello Morning World");
}
```

Python 概説

■ 2020年10月 人気度：TIOBE



Oct 2020	Oct 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.95%	+0.77%
2	1	▼	Java	12.56%	-4.32%
3	**3**		**Python**	**11.28%**	**+2.19%**
4	4		C++	6.94%	+0.71%

AI（機械学習，統計，分析）
での利用が起因？
- 第三次 AIブーム（2006年?～）
- 非プログラマー

・習得の容易性
・豊富なライブラリー
・活用範囲の広さ

2018年ごろから急激な増加
→ Java に変わる言語として注目か

※1) TIOBE のソース
条件を満たした 25の検索エンジンのヒット数
(詳細は以下のリンクを参照)

https://www.tiobe.com/tiobe-index/

https://www.tiobe.com/tiobe-index/programming-languages-definition/

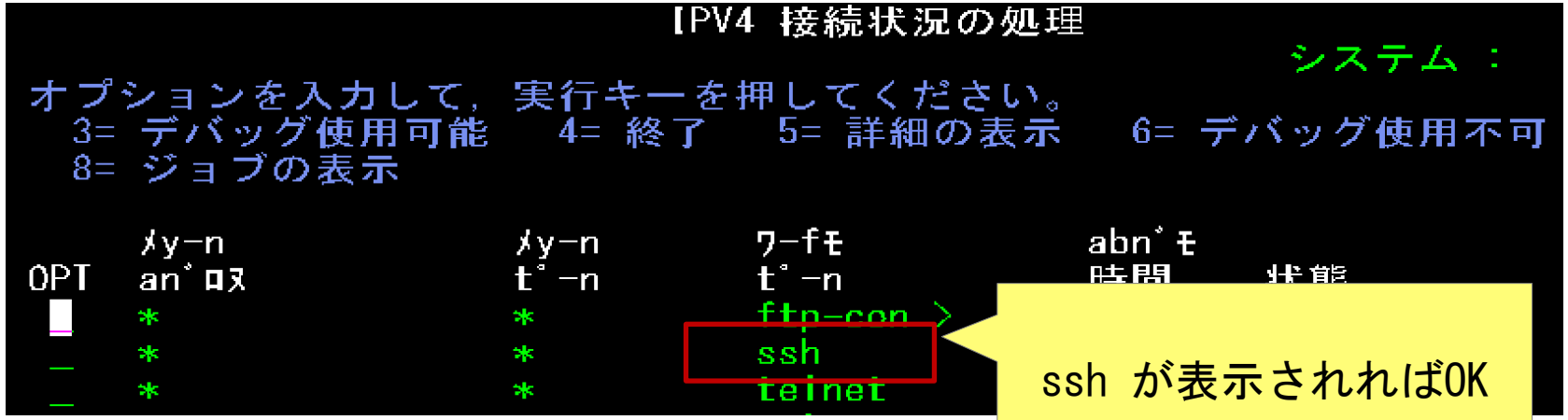
Python のインストール

インストール

- 流れ
  1. sshd の開始
  2. yum を導入
  3. Python の導入

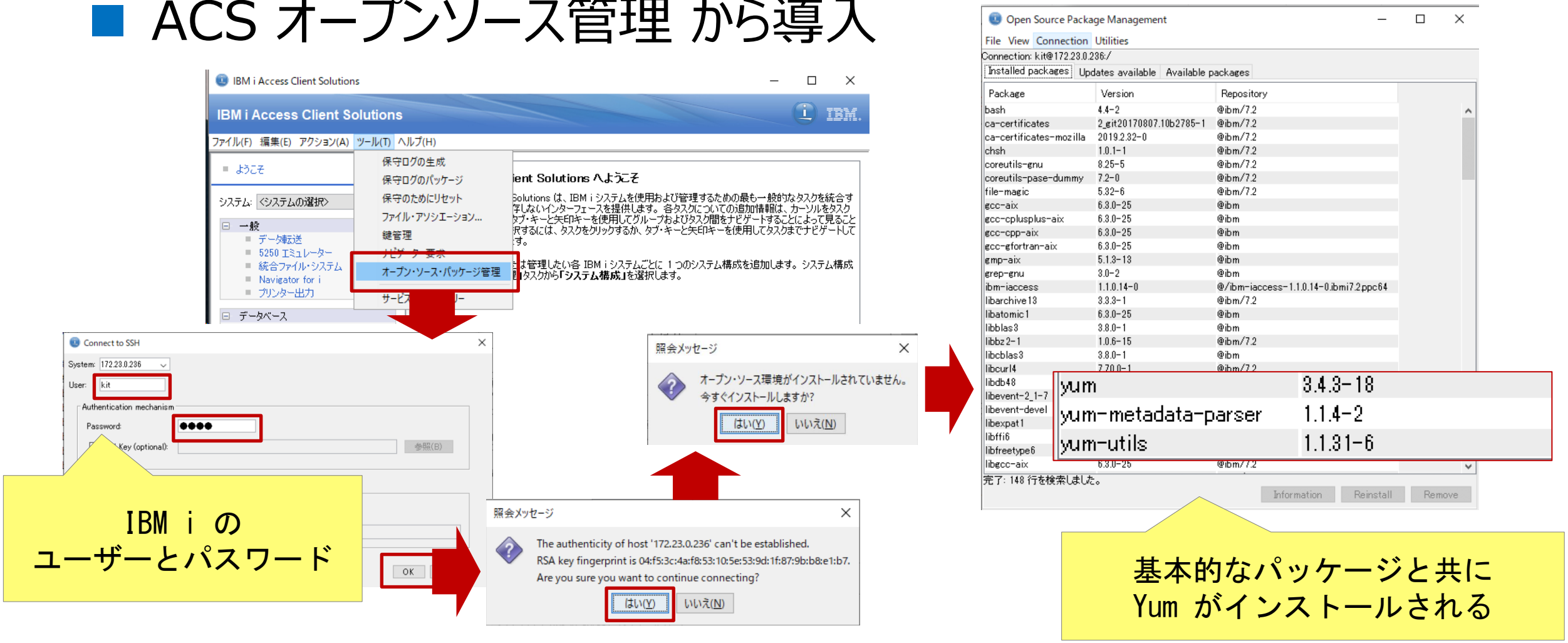
インストール：1. sshd の開始

- 5250 を利用
  - 5250エミュレーターで以下の CL コマンドを実行
    - STRTCPSVR SERVER(*SSHD)
  - 開始は以下 の CL コマンドで確認
    - NETSTAT OPTION(*CNN)



インストール：2. yum の導入

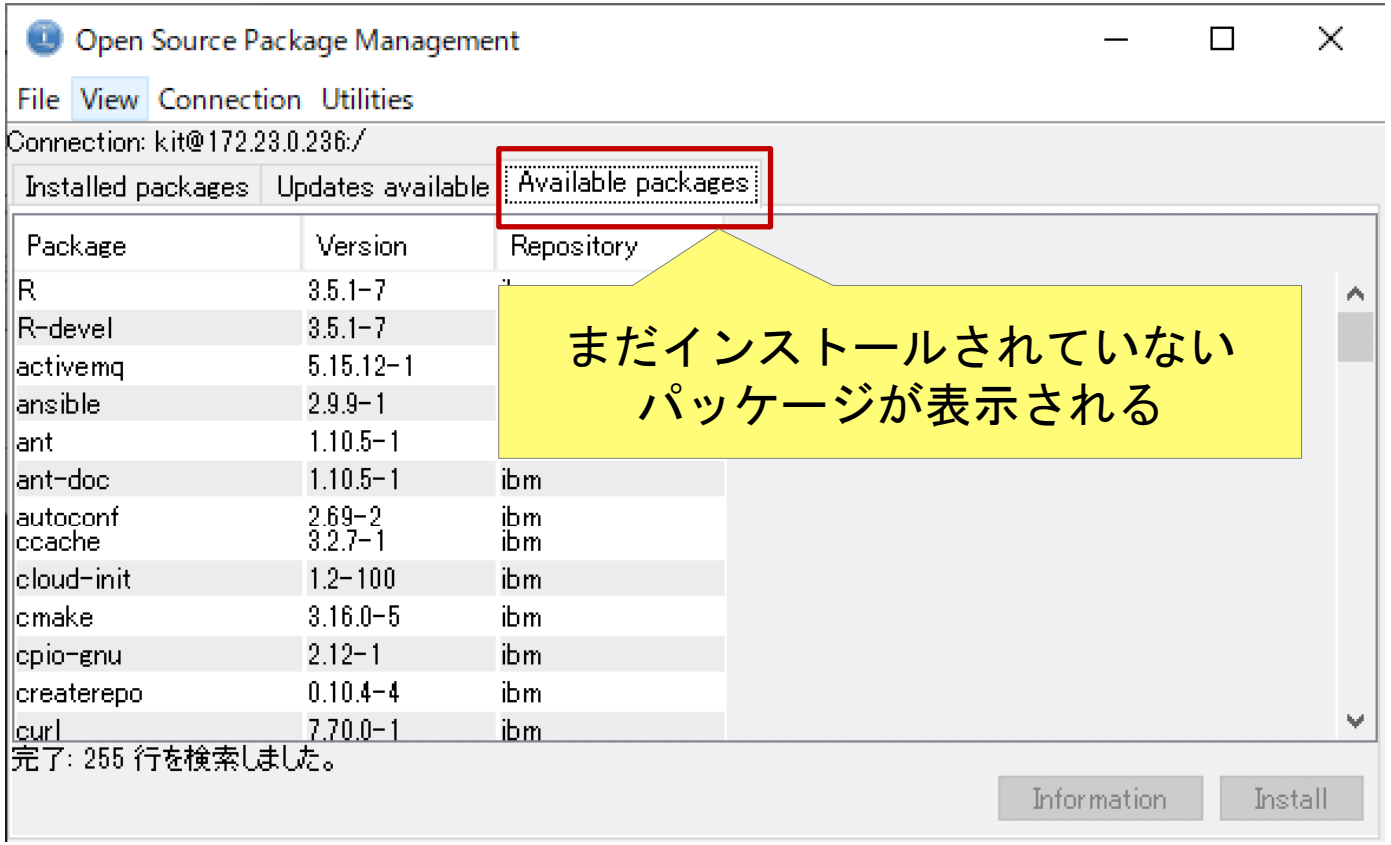
- ACS オープンソース管理 から導入



※ 詳細は yum 分科会の発表資料をご参照ください。 https://i5php.jp/wp-content/uploads/2019/09/yum_guide.pdf

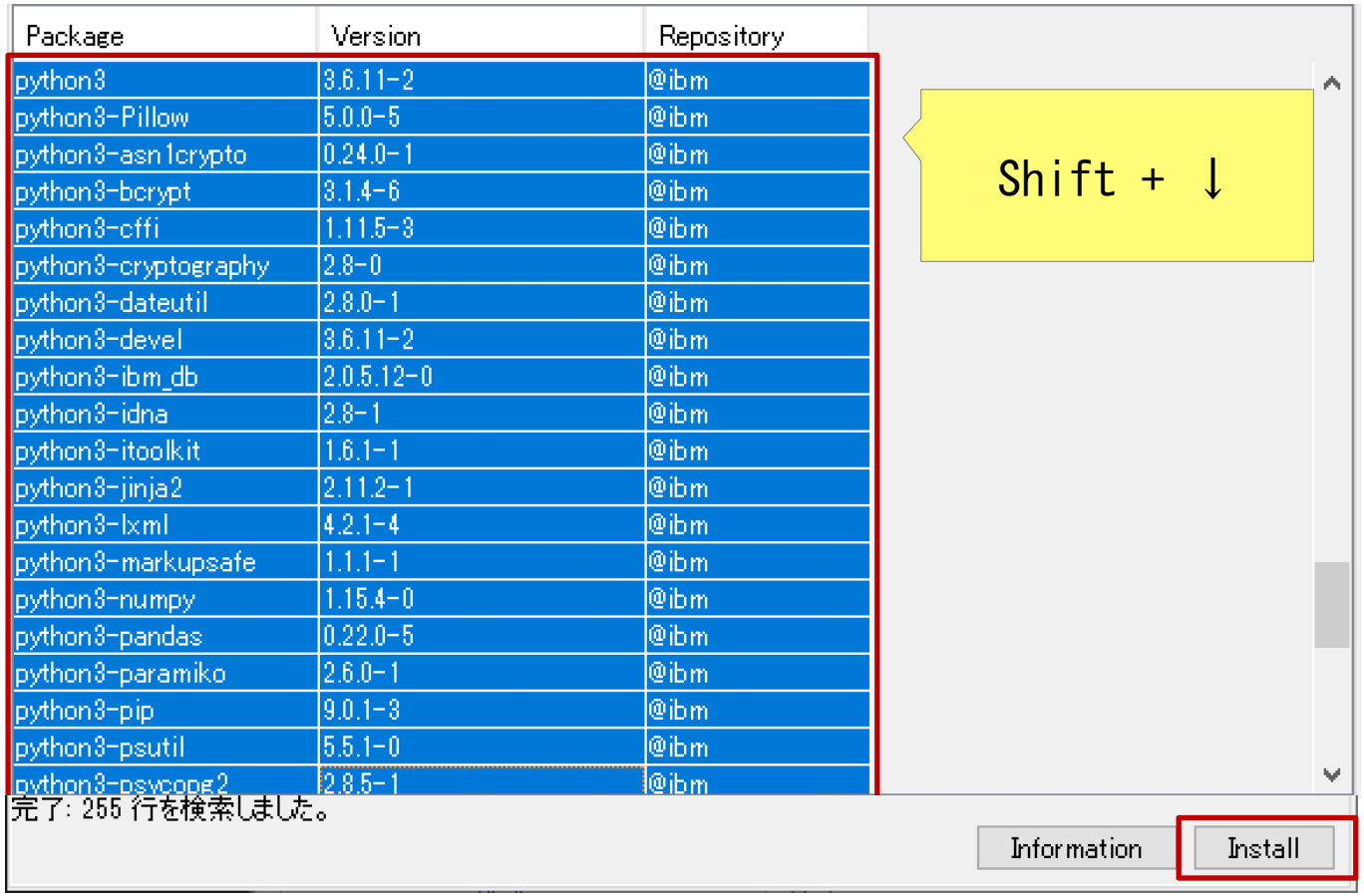
インストール：3. Python の導入 1

- ACS オープンソース管理 から導入
  1. Open Source Package Management の「Available Packages」タブを開く



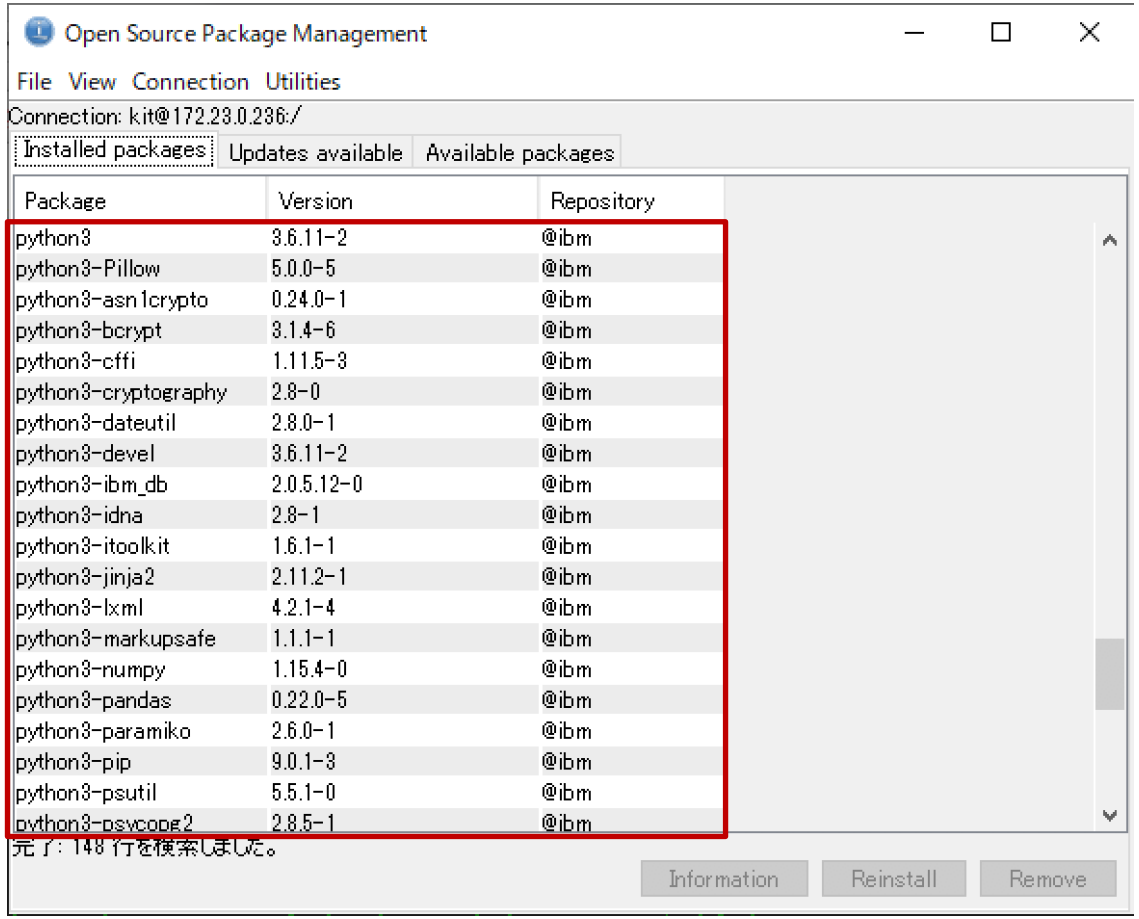
インストール：3. Python の導入 2

- ACS オープンソース管理 から導入
  2. Python3 で始まるパッケージを全て選択し「Install」



インストール：3. Python の導入 2

- ACS オープンソース管理 から導入
  3. Install package タブに表示されればOK



yum コマンド の場合
$ yum install python3*

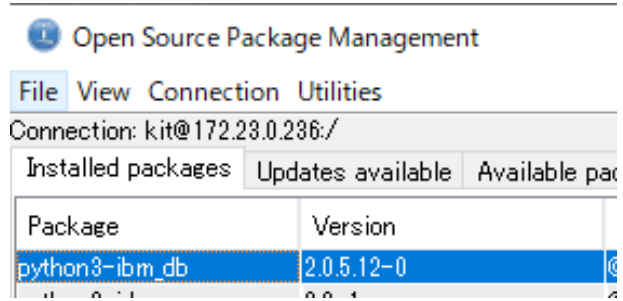
データベース・アクセス

データベース・アクセス

- Db2 for i へのアクセス方法は 2種類
  - **ibm_db**  今回はこちら
  - **pyODBC**

データベース・アクセス：ibm_db

- インストール
  - ACS の場合
    - ACS で python3-ibm_db を選択し Install
  - yum コマンドの場合
    - yum install python3-ibm_db



- 2種類のモジュールが含まれる
  - **ibm_db** ：ベース・モジュール
    - SQL 照会の発行、ストアード・プロシージャーの呼び出し等
  - **ibm_db_dbi** ： Python データアクセス仕様に準拠
    - PEP249：Python Database API Specification v2.0

今回はこちら

データベース･アクセス：shell で実行してみた

■ ibm_db_dbi の利用

exdb01_app_cli.py

```
1  import ibm_db_dbi as db
2  import io,sys
3  sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
4
5  conn = db.connect("DATABASE=*LOCAL;","KIT","<password>")
6  cursor = conn.cursor()
7  cursor.execute("SELECT * FROM QEOL.TOKMSP")
8  for r in cursor.fetchall():
9      print(r[0], end=' ')
10     print(r[1], end=' ')
11     print(r[2])
```

$ python exdb01_app_cli.py

データベース･アクセス: shell で実行してみた

解説

```
1  import ibm_db_dbi as db
2  import io,sys
3  sys.stdout = io.TextIOWrapper(sys.stdout.buffer, encoding='utf-8')
4
5  conn = db.connect("DATABASE=*LOCAL;","KIT","<password>")
6  cursor = conn.cursor()
7  cursor.execute("SELECT * FROM QEOL.TOKMSP")
8  for r in cursor.fetchall():
9      print(r[0], end=' ')
10     print(r[1], end=' ')
11     print(r[2])
```

ibm_db_dbi
モジュールの読み込み

標準出力を utf-8 に設定しないと
UnicodeEncodeError になる

IBM I との接続
リレーショナル
データベース名を
指定

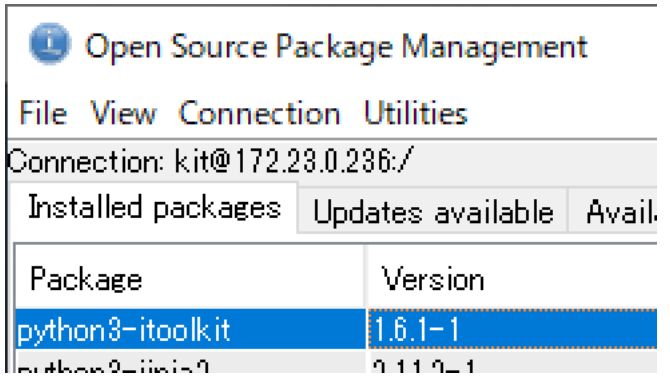
カーソル作成と
SQLの実行

フェッチし、ループ
レコード1件毎に
出力

itoolkit

itoolkit

- インストール
  - ACS の場合
    - ACS で python3-itoolkit を選択し Install
  - yum コマンドの場合
    - yum install python3-itoolkit



itoolkit：shell で実行してみた

itoolkit の利用

exxt01_app_cli.py

```python
import ibm_db_dbi
from itoolkit import *
from itoolkit.transport import DatabaseTransport

conn = ibm_db_dbi.connect("DATABASE=*LOCAL","KIT","<password>")
itransport = DatabaseTransport(conn, ctl="*pase(1208/5035)")

itool = iToolKit()
itool.add(iCmd5250('dspsyssts', 'DSPSYSSTS'))
itool.call(itransport)
result = itool.dict_out('dspsyssts')
print(result['dspsyssts'])
```

```
$ export LANG=JA_JP.UTF-8
$ python exxt01_app_cli.py
```

T&Trust Tenderness and Trust

itoolkit：shell で実行してみた

解説

```
1  import ibm_db_dbi
2  from itoolkit import *
3  from itoolkit.transport import DatabaseTransport
4
5  conn = ibm_db_dbi.connect("DATABASE=*LOCAL","KIT","<password>")
6  itransport = DatabaseTransport(conn, ctl="*pase(1208/5035)")
7
8  itool = iToolKit()
9  itool.add(iCmd5250('dspsyssts', 'DSPSYSSTS'))
10 itool.call(itransport)
11 result = itool.dict_out('dspsyssts')
12 print(result['dspsyssts'])
13
```

ibm_db_dbi,itookit
モジュールの読み込み

IBM i との接続

コマンド実行時の設定

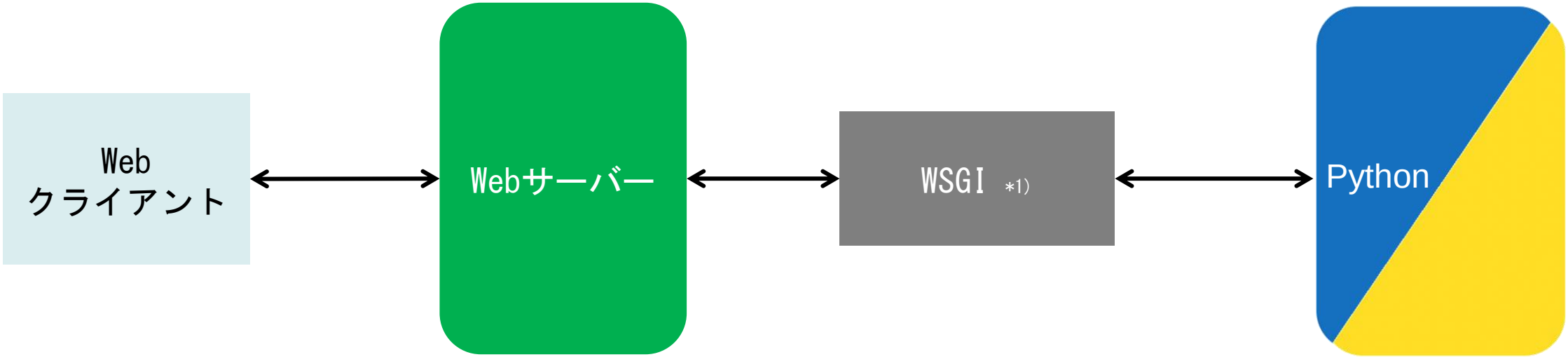
コマンドの指定と実行

結果の出力

Web サーバー連携

概要

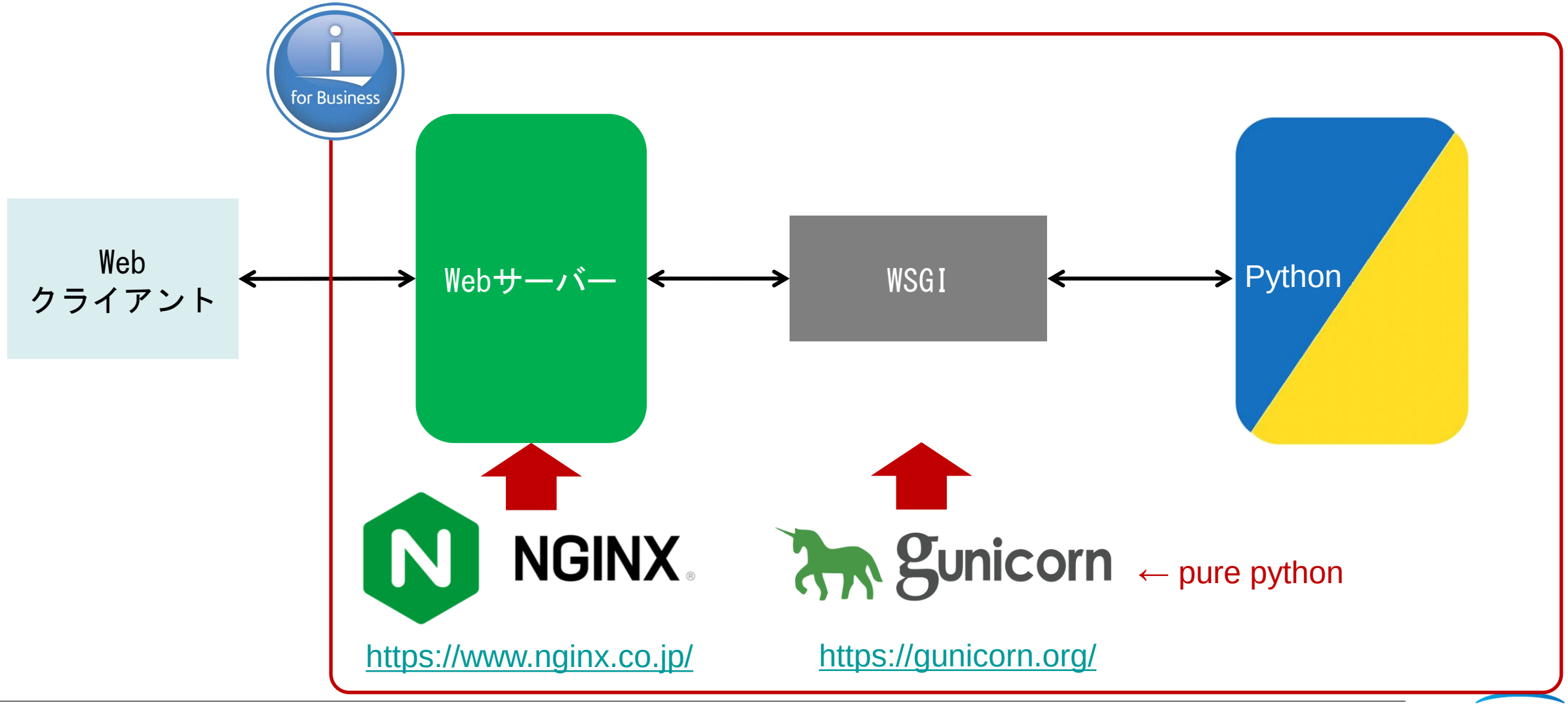
Python Web アプリケーション構成



*1) Web Server Gateway Interface

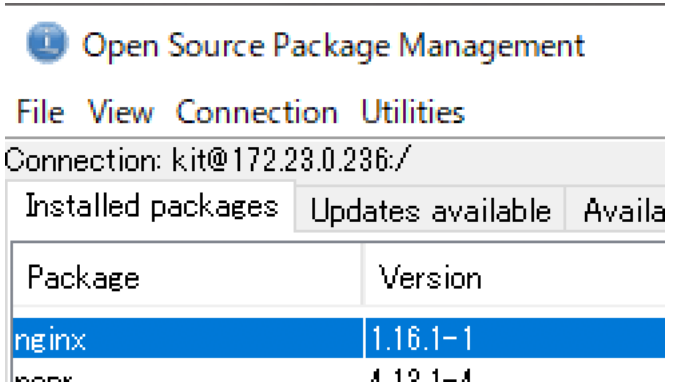
Web サーバー連携

- 今回試した構成



Web サーバー連携

- Nginx インストール
  - ACS の場合
    - ACS で nginx を選択し Install
  - yum コマンドの場合
    - yum install nginx



- Gunicorn インストール
  - pip コマンド *2)
    - pip install gunicorn

*2) Pip Install Package : Python のパッケージ管理システム

```
bash-4.4$ pip install gunicorn
Collecting gunicorn
  Downloading https://files.pythonhosted.org/packages/69/ca/926f7cd3a2014b16870086b2d0fdc84a9e49473c68a8dff8b57f7c156f43/gunicorn-20.0.4-py2.py3-none-any.whl (77kB)
    100% |################################| 81kB 1.2MB/s
Requirement already satisfied: setuptools>=3.0 in /QOpenSys/pkgs/lib/python3.6/site-packages (from gunicorn)
Installing collected packages: gunicorn
Successfully installed gunicorn-20.0.4
```

Web サーバー連携：参考

nginx.conf

```
bash-4.4$ cat nginx.conf
worker_processes  1;

events {
    worker_connections 512;
}

http {
    server {
        listen  9123;
        server_name INFRA-PRACTICE-NGINX;
        charset UTF-8;

        proxy_set_header    Host    $host;

        location / {
            proxy_pass http://127.0.0.1:9876;
        }
    }
}
```

gunicorn_settings.py

```
bash-4.4$ cat gunicorn_settings.py
import os

bind = '127.0.0.1:' + str(os.getenv('PORT', 9876))
proc_name = 'Infrastructure-Practice'
workers = 1
```

配置

```
bash-4.4$ tree -L 2 /QOpenSys/usr/share/nginx_gp
/QOpenSys/usr/share/nginx_gp
├── config
│   ├── __pycache__
│   ├── gunicorn_settings.py
│   └── nginx.conf
```

Web サーバー連携：データベース･アクセス

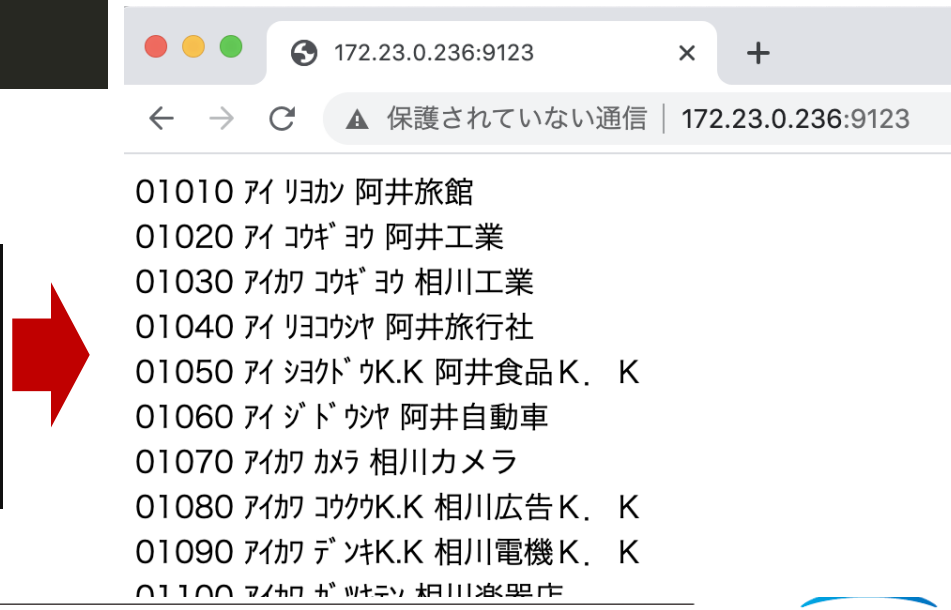
exdb01_app.py

```python
import ibm_db_dbi as db

def application(env, start_response):
    start_response('200 OK', [('Content-Type', 'text/html')])
    conn = db.connect("DATABASE=*LOCAL","KIT","KITA")
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM QEOL.TOKMSP")
    ret = ""
    for r in cursor.fetchall():
        ret += r[0] + ' ' + r[1] + ' ' + r[2] + '</br>'
    return [bytes(ret,'utf-8')]

```

```
bash-4.4$ tree -L 2 /QOpenSys/usr/share/nginx_gp
/QOpenSys/usr/share/nginx_gp
├── config
│   ├── __pycache__
│   ├── gunicorn_settings.py
│   └── nginx.conf
├── exdb01_app.py
```

Nginx, Gunicorn 起動

```
bash-4.4$ nginx -c /QOpenSys/usr/share/nginx_gp/config/nginx.conf
bash-4.4$ cd /QOpenSys/usr/share/nginx_gp
bash-4.4$ gunicorn exdb01_app:application -c config/gunicorn_settings.py --reload
[2020-10-16 21:25:42 JST] [4848] [INFO] Starting gunicorn 20.0.4
[2020-10-16 21:25:42 JST] [4848] [INFO] Listening at: http://127.0.0.1:9876 (4848)
[2020-10-16 21:25:42 JST] [4848] [INFO] Using worker: sync
[2020-10-16 21:25:42 JST] [4850] [INFO] Booting worker with pid: 4850
```



Web サーバー連携：itoolkit

exxt01_app.py

```python
from itoolkit import *
from itoolkit.transport import DatabaseTransport
import ibm_db_dbi

def application(env, start_response):
    start_response('200 OK', [('Content-Type', 'text/plain')])
    conn = ibm_db_dbi.connect("DATABASE=*LOCAL","KIT","KITA")
    itransport = DatabaseTransport(conn,ctl="*pase(1208/5035)")

    itool = iToolKit()
    itool.add(iCmd5250('dspsyssts', 'DSPSYSSTS'))
    itool.call(itransport)
    result = itool.dict_out('dspsyssts')
    return [bytes(str(result['dspsyssts']),'utf-8')]
```

```
bash-4.4$ tree -L 2 /QOpenSys/usr/share/nginx_gp
/QOpenSys/usr/share/nginx_gp
├── config
│   ├── __pycache__
│   ├── gunicorn_settings.py
│   └── nginx.conf
└── exxt01_app.py
```

Nginx, Gunicorn 起動

```
bash-4.4$ nginx -c /QOpenSys/usr/share/nginx_gp/config/nginx.conf
bash-4.4$ cd /QOpenSys/usr/share/nginx_gp
bash-4.4$ gunicorn exxt01_app:application -c config/gunicorn_settings.py --reload
[2020-10-16 22:01:22 JST] [4914] [INFO] Starting gunicorn 20.0.4
[2020-10-16 22:01:22 JST] [4914] [INFO] Listening at: http://127.0.0.1:9876 (4914)
[2020-10-16 22:01:22 JST] [4914] [INFO] Using worker: sync
[2020-10-16 22:01:22 JST] [4916] [INFO] Booting worker with pid: 4916
```

ZendPHP7（ZendServer）

PHP と比較

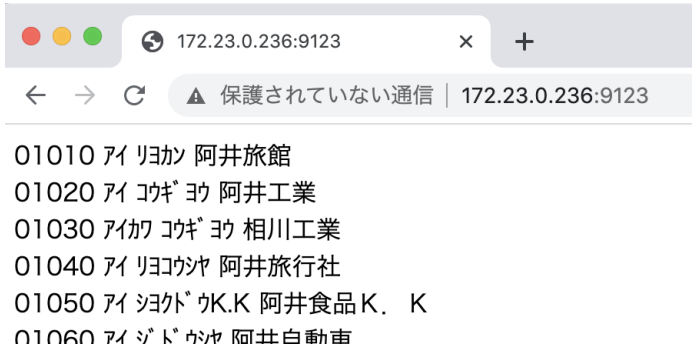
PHP との比較：データベース・アクセス

Python

```python
import ibm_db_dbi as db

def application(env, start_response):
    start_response('200 OK', [('Content-Type', 'text/html')])
    conn = db.connect("DATABASE=*LOCAL","KIT","KITA")
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM QEOL.TOKMSP")
    ret = ""
    for r in cursor.fetchall():
        ret += r[0] + ' ' + r[1] + ' ' + r[2] + '</br>'
    return [bytes(ret,'utf-8')]

```

PHP

```php
<?php

header("Content-type: text/html; charset=utf-8");
$conn = db2_connect('*LOCAL', 'KIT', 'KITA');
$result = db2_exec($conn, "SELECT * FROM QEOL.TOKMSP");
$ret = "";
while($r = db2_fetch_array($result)) {
  $ret .= $r[0]." ".$r[1]." ".$r[2]."</br>";
}
echo $ret;

?>
```



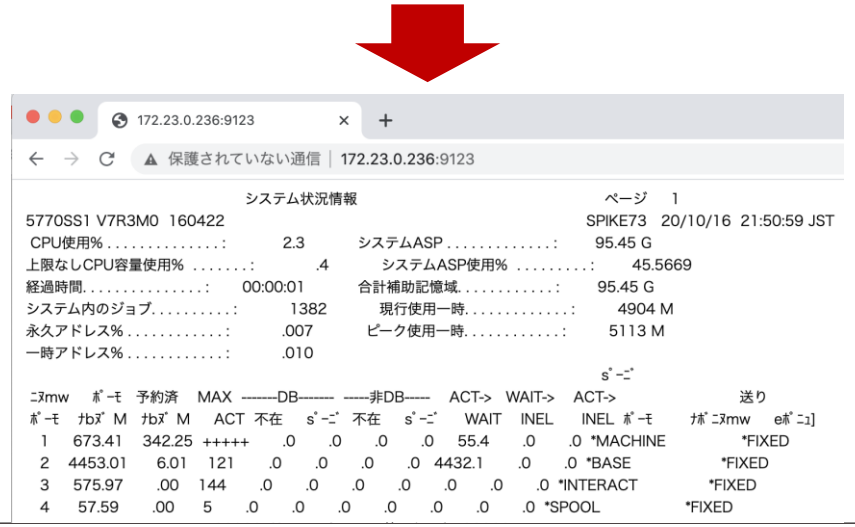
PHP との比較：TOOLKIT

Python

```python
from itoolkit import *
from itoolkit.transport import DatabaseTransport
import ibm_db_dbi

def application(env, start_response):
    start_response('200 OK', [('Content-Type', 'text/plain')])
    conn = ibm_db_dbi.connect("DATABASE=*LOCAL","KIT","KITA")
    itransport = DatabaseTransport(conn,ctl="*pase(1208/5035)")

    itool = iToolKit()
    itool.add(iCmd5250('dspsyssts', 'DSPSYSSTS'))
    itool.call(itransport)
    result = itool.dict_out('dspsyssts')
    return [bytes(str(result['dspsyssts']),'utf-8')]
```

PHP

```php
<?php
include_once 'ToolkitService.php';

header("Content-type: text/plain; charset=utf-8");
$conn = db2_connect('*LOCAL', 'KIT', 'KITA');
$itool = ToolkitService::getInstance($conn);

$result = $itool->CLInteractiveCommand("DSPSYSSTS");
$ret = "";
foreach($result as $r)
  $ret .= $r."\n";
echo $ret;
?>
```



今後の Python 分科会活動

Python 分科会の活動候補

■ 今後の Python 検証

- 今回の続き
  - pyODBC, ストアドプロシージャ（RPG連携）, iToolkit, PHPとの比較
- Web アプリケーション
  - フレームワーク(Django/Flask), パフォーマンス, Rest API
- 解析
  - IBM i だけで「画像解析」がどこまでできるのか
  - RFE に OpenCV をリクエスト中です。投票 お願いします。
    - https://www.ibm.com/developerworks/rfe/execute?use_case=viewRfe&CR_ID=140124
- 機械学習
  - IBM i だけで何が、どこまでできるのか

※その他、検証して欲しい事があればアンケートにご記入ください

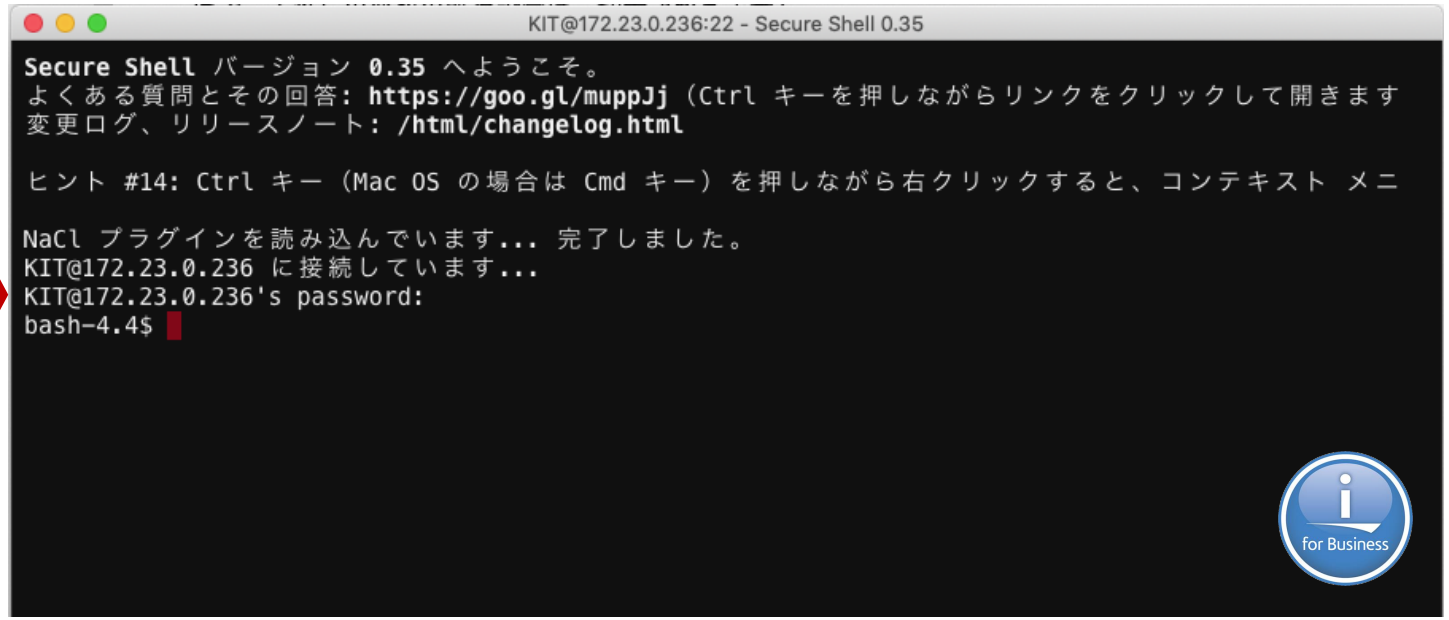
IBM i の pase 環境を利用するにあたり、今回は chrome secure shell を利用しました

準備：Chrome secure shell

準備：Chrome Secure Shell

- Chrome Secure Shell の利用
  1. Chrome ブラウザで「chrome ウェブストア」を表示
     - https://chrome.google.com/webstore
  2. 「ストアを検索」に “secure shell app” と入力し Enter
  3. 一覧から “Secure Shell” (以下のアイコン)をクリック
  4. 「Chrome に追加」をクリック
  5. 確認画面で「拡張機能を追加」をクリック
  6. Chrome の右上にある「拡張機能アイコン」をクリック
  7. 表示された「接続ダイアログ」をクリック
  8. ユーザー、IBM i の IPアドレス、ポート(22)を入力し [ENTER]接続
  9. 接続後、パスワードを入力（初回のみ接続確認があるので yes で回答）

準備：Chrome Secure Shell



事前に以下を対応しておくと使いやすいです。
※ 5250 から実施

1. /home/<ユーザー>を作成
2. .profile 作成
3. .bashrc 作成

```
bash-4.4$ pwd
/home/kit
bash-4.4$ cat .profile
bash
bash-4.4$ cat .bashrc
alias python=python3
alias pip=pip3
export PATH=/QOpenSys/pkgs/bin:/QOpenSys/pkgs/sbin:$PATH
```