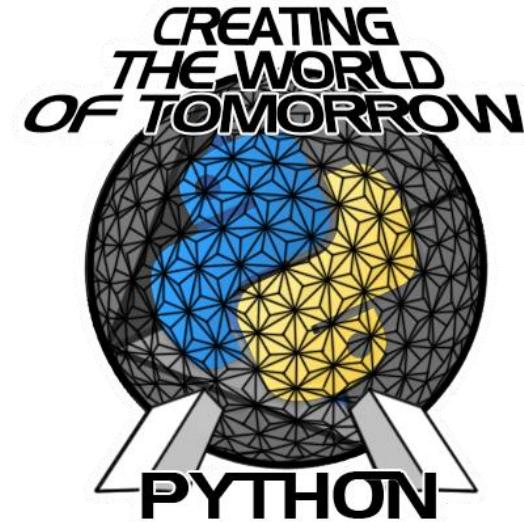


Pragmatic Python for Pioneering Programmers

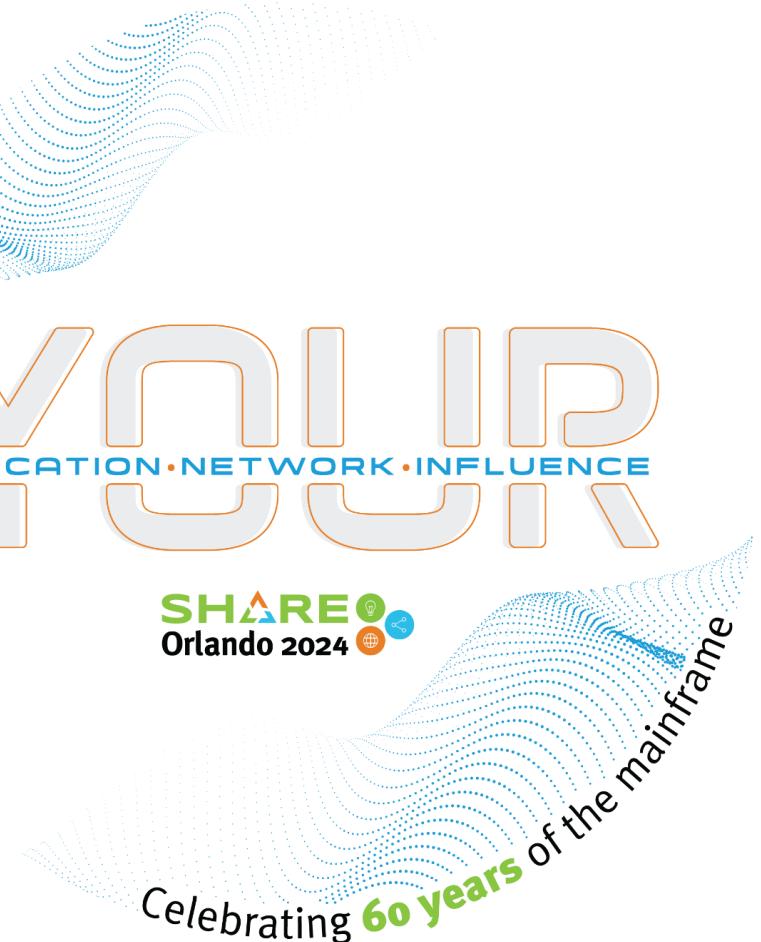


Michael Fontanetta, Ensono
Steven Perva, Ensono

Celebrating **60 years** of the mainframe



SHARE
Orlando 2024



Scripting Sidekicks – Your Trusty Navigators

Dr. Stephen Peru

STEVEN PERVA

STRENGTHS:

- aplomb
- Generalist
- Networking
- T-Shirts

WEAKNESSES:

- Jira



Prof. Marcel Fontana

MICHAEL FONTANETTA

STRENGTHS:

- Vision
- Analytics
- Herding Kewl Cats
- Adaptation

WEAKNESSES:

- Overly Ambitious
- Cake

Your ally for better outcomes

300+
clients

\$900M
2023 revenue target

11+ years
average client tenure

Why clients choose us:



Expertise



Flexibility

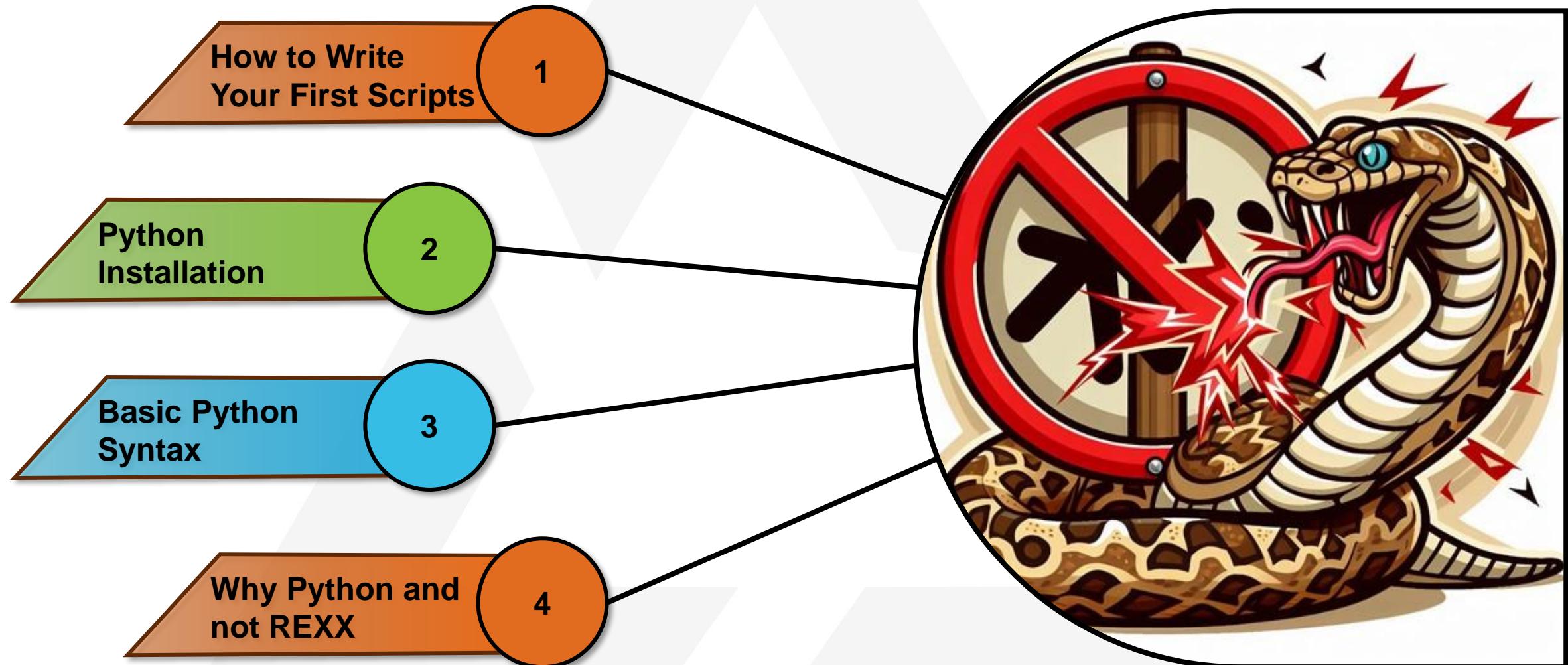


Culture

"Ensono has experience operating in a **very complex environment - everything from mainframe to cloud. They are a real ally...so easy to do business with."**

- Phil Ratcliff, CEO illumifin

What We're Skipping



What We're Diving Into

ZOAU as a
Python Package

1

How to Retrieve
Packages

2

Scripting
Examples

3

Common Issues

4

Operationalizing
Python Scripts

5



A Pythons View of ZOAU

Bridge Between Python and z/OS

Running Operator Commands

Running z/OS Load Modules



ZOAU Operator Commands



`zooutil_py.opercmd.execute(command, *args, **kwargs)`

```
from zooutil_py import opercmd

resp = opercmd.execute('DISPLAY IPLINFO')

if resp.rc == 0:
    print(resp.stdout_response)
else:
    print(resp.stderr_response)
```

REFERENCE:

https://www.ibm.com/docs/en/zoau/1.3.x?topic=SSKFYE_1.3.x/zoau-python-opercmd.htm

ZOAU Operator Commands



LPAR	2024057 11:55:55.00	ISF031I CONSOLE TSSX0000 ACTIVATED
LPAR	2024057 11:55:55.00	-DISPLAY IPLINFO
LPAR	2024057 11:55:55.00	IEE254I 11.55.55 IPLINFO DISPLAY 401 SYSTEM IPLED AT 14.09.17 ON 12/28/2023 RELEASE z/OS 02.05.00 LICENSE = z/OS USED LOAD00 IN SYS1.IPLPARM ON 01003 ARCHLVL = 2 MTLSHARE = N VALIDATED BOOT: NO IEASYM LIST = (R0,L) IEASYS LIST = R0 (OP) IODF DEVICE: ORIGINAL(01003) CURRENT(01003) IPL DEVICE: ORIGINAL(01112) CURRENT(01112) VOLUME(LPARA2)

ZOAU Operator Commands

```
IBMUSER:/u/IBMUSER/scripts: >python display_iplinfo.py
{'ipl_time': '14.09.17',
 'ipl_date': '12/28/2023',
 'fmid': 'z/OS 02.05.00',
 'system': 'z/OS',
 'loadxx_member': 'LOAD00',
 'loadxx_dsn': 'SYS1.IPLPARM',
 'loadxx_device': '01003',
 'archlvl': '2',
 'mtlshare': 'N',
 'validated_boot': 'NO',
 'ieasym_list': [ 'R0' ],
 'ieasys_operator_provided': True,
 'ieasys_list': [ '00', 'R0' ],
 'iodf_device_original': '01003',
 'iodf_device_current': '01003',
 'ipl_device_original': '01112',
 'ipl_device_current': '01112'}
```

ZOAU Load Modules

```
zooutil_py.mvscmd.execute(pgm, pgm_args=None, dds=[], *args, **kwargs)  
zooutil_py.mvscmd.executeAuthorized(pgm,  
                                     pgm_args=None, dds=[], *args, **kwargs)
```

```
from zooutil_py import mvscmd  
  
result = mvscmd.execute("IEFBR14")  
if result.rc == 0:  
    print(result)
```



REFERENCE:

https://www.ibm.com/docs/en/zau/1.3.x?topic=SSKFYE_1.3.x/zau-python-mvscmd.htm

ZOAU Load Modules

```
from zooutil_py import datasets, mvscmd
from zooutil_py.types import DDStatement, DatasetDefinition

tmp_name = datasets.tmp_name(hlq="IBMUSER")
tmp_ds = datasets.create(tmp_name, "SEQ", primary_space = "1K",
                        record_format="FB", record_length=256)
dd_statements = [
    DDStatement(name="SYSOUT", definition=DatasetDefinition(tmp_name))
]
result = mvscmd.executeAuthorized("HZSPRNT",
                                  pgm_args="CHECK(IBMSS,USS_PARMLIB)",
                                  dds=dd_statements)
if result.rc == 0:
    output = datasets.read(tmp_name)
    [print(line.strip()) for line in output.split('\n')]
```

ZOAU Load Modules

```
IBMUSER :/u/IBMUSER/scripts: >python hc_detail.py
*****
* HZSPRINT (HBB77C0-19034) 2024/02/26 15:46 *
* HZSU001I Check messages *
* Sysplex: LPRPLEX System: LPAR *
* Filter: CHECK(IBMUSS,USS_PARMLIB) *
*****  

*****  

* Start: CHECK(IBMUSS,USS_PARMLIB) *
*****  

*****  

CHECK(IBMUSS,USS_PARMLIB)
SYSPLEX: LPRPLEX SYSTEM: LPAR
START TIME: 02/26/2024 15:10:58.632666
CHECK DATE: 20060112 CHECK SEVERITY: LOW
```

Harnessing the Power of Packages



Why Packages?

- Enhanced Functionality
- Reusable Code
- Highly Performant
- Continuous Improvement

How to Get Packages?

- Compiler Setup
- Determine Risk Profile
- Implement Package Process

Setting Up a Compiler

You can use:

- IBM C/C++ For Open Enterprise Languages on z/OS 2.0
- IBM Open XL C/C++ 1.1 for z/OS
- IBM XL C/C++ 2.4.1 for z/OS 2.4 and z/OS 2.5
- IBM z/OS XL C/C++

Recommended to use:

- IBM C/C++ For Open Enterprise Languages on z/OS® 2.0
- IBM Open XL C/C++ 1.1

While Python is Interpreted, some packages require compilation

Python packages can leverage lower-level languages

Performance of low level with the ease of use of Python

Performance Optimizations

Native Code for S/390 (s390) Architecture

Not required for precompiled packages.

Not required for pure Python packages.

Most can get by without a compiler setup.

REFERENCE:

<https://www.ibm.com/docs/en/python-zos/3.12?topic=using-cc-compilers-open-enterprise-sdk-python-312>



Secure Packages for All!



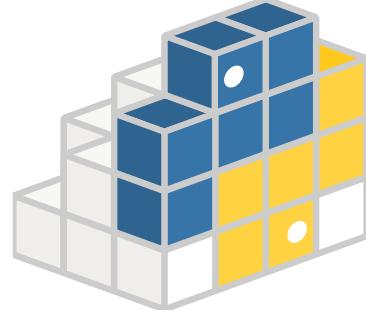
- **Secured Connectivity**

- Packages haven't been tampered with
- Watering Hole or Supply Chain Attacks
- Regulations and compliance

- **Trivial attack: Common typos in the name of the package**

- requests instead of requests
- pandass instead of pandas

Choosing an Alternative Repository



- IBM Vetted Security of Packages
- Purpose-built for s390x Architecture (z/OS)
- Easy to Use Web Interface
- Free (with Optional Support)



- Compliance
- Internal-Only / “Offline” Access
- Support for Multiple Technologies
(Java, npm, NuGet, Docker, etc.)
- Proxy and Cache



REFERENCE:

<https://www.redbooks.ibm.com/redpapers/pdfs/redp5709.pdf>

The Masochist's Approach



- On a system **with internet access**:
 - Create a requirements.txt file:

```
package1==1.0
package2==2.1.0
```
 - Issue `pip download -r requirements.txt`
 - Transfer requirements.txt and downloaded packages to system without internet access
 - You **will want** to use `--platform`, `--python-version`, and `--only-binary` switches
- On your LPAR **without internet access**:
`pip install -no-index -find-links <path where packages live> -r requirements.txt`
 - We recommend using either the AI Toolkit or an internal Nexus Repository when available. If you must do manual, use an alternate repository instead of public.

REFERENCE:

https://pip.pypa.io/en/stable/cli/pip_download/

About that ZOAU Package

- ZOAU Ships with a wheel file for easy API enablement
- Located in \$ZOAU_HOME as
`zooutil_py-<version>-cp<python_version>-none-any.whl`
- `pip install $ZOAU_HOME/<wheel_file>`
- e.g. `pip install \$ZOAU_HOME/zooutil_py-1.2.5.0-cp311-none-any.whl`
- **Test it:** `python -c "import zooutil_py"`
- Should return nothing



What a Sysprog Thinks is Cool

```
IBMUSER :/u/IBMUSER/scripts: >python display_iplinfo.py
{'ipl_time': '14.09.17',
 'ipl_date': '12/28/2023',
 'fmid': 'z/OS 02.05.00',
 'system': 'z/OS',
 'loadxx_member': 'LOAD00',
 'loadxx_dsn': 'SYS1.IPLPARM',
 'loadxx_device': '01003',
 'archlvl': '2',
 'mtlshare': 'N',
 'validated_boot': 'NO',
 'ieasym_list': [ 'R0' ],
 'ieasyoperator_provided': True,
 'ieasylist': ['00', 'R0'],
 'iodf_device_original': '01003',
 'iodf_device_current': '01003',
 'ipl_device_original': '01112',
 'ipl_device_current': '01112'}
```

What a Sysprog Thinks is Cool

```
IBMUSER :/u/IBMUSER/scripts: >python display_parmlib.py
{'source': 'IPL',
 'entries': [{ 'order': '1',
               'specified': 'S',
               'volume': 'LPARIX',
               'dsn': 'OEML.PARMLIB.LPAR'},
              { 'order': '2',
               'specified': 'S',
               'volume': 'LPARA2',
               'dsn': 'SYS1.IBM.PARMLIB'},
              { 'order': '3',
               'specified': 'S',
               'volume': 'LPARM1',
               'dsn': 'SYS1.PARMLIB'}]}
```

What a Sysprog Thinks is Cool

1. Using display_iplinfo, get the IEASYS suffixes.
2. Using display_parmlib, get the PARMLIB concatenation.
3. Using both modules, we can read every IEASYSxx member, build one large IEASYSxx buffer.
4. Programmatically run through the buffer and find the SMFPRMxx (SMF=) dataset in the PARMLIB.

```
IBMUSER :/u/IBMUSER/scripts: >python locate_smfprmxx.py
{'entries': ['OEML.PARMLIB.LPAR(SMFPRMR0)']}
```

5. **TBD** Serialize SMFPRMxx to log or make intelligent decisions based on the SMF configuration.

What a Sysprog Thinks is Cool

```
IBMUSER :/u/IBMUSER/scripts: >python syslog_serializer.py
{
    "Record Type": "Single-Line Message",
    "Record Source": "No Source",
    "Routing Codes": "40000000",
    "System Name": "LPAR",
    "Julian Date": "24064",
    "Time Stamp": "05:33:35.55",
    "Identifier": "TSU05751",
    "Message Flags": [
        "Message Not Serviced by User Exit",
        "Automation Requested"
    ],
    "Message Text": " IGD104I IBMUSER.LPAR.SPFTEMP1.CNTL RETAINED,\n"
}
```

```
IBMUSER:/service: > curl -k
https://public.dhe.ibm.com/ibmdl/export/pub/software/htp/zos/tools/zoau/zoau-1.2.5.0.pax.Z
-o zoau-1.2.5.0.pax.Z
% Total    % Received % Xferd  Average Speed   Time      Time      Time  Current
                                         Dload  Upload   Total   Spent    Left  Speed
100 4914k  100 4914k     0      0  2999k      0  0:00:01  0:00:01  --:--:-- 3001k
```

```
IBMUSER:/u/IBMUSER/installers: >python zoau.py /service/zoau-1.2.5.0.pax.Z
IBMUSER.ZOAU.ZFS
ZOAU Installation Script 0.1.1
Installing from /service/zoau-1.2.5.0.pax.Z...
Detected version v1.2.5.0 from archive...
Allocating filesystem IBMUSER.ZOAU.ZFS...
Mounting filesystem IBMUSER.ZOAU.ZFS to /tmp/tmp1hjyr_aj...
Unpacking archive to filesystem...
Unmounting and cleaning up temporary mountpoint...
Checking target installation mountpoint...
Path /usr/lpp/IBM/zoautil exists.
Path /usr/lpp/IBM/zoautil is not empty. Verifying if path is a mountpoint...
Path /usr/lpp/IBM/zoautil is a mountpoint.
PRODUCTS.ZOAU.ZFS mounted to "/usr/lpp/IBM/zoautil". Unmount now? [Y/N] (Y) : y
Unmounting PRODUCTS.ZOAU.ZFS...
PRODUCTS.ZOAU.ZFS unmounted. Mounting IBMUSER.ZOAU.ZFS...
IBMUSER.ZOAU.ZFS successfully mounted...
Exiting...
IBMUSER:/u/IBMUSER/installers: >zoaversion
2023/09/20 17:20:50 CUT V1.2.5.0 dfc0ed10 3938 PH54246 998 7544c1ea
```

A Directors Perspective

- Solving Front-Office Problems
- How to Identify Target Problems
- Leverage New Talent
 - Interns
 - Junior Staff



SCRT Process



- **The Problem:**
 - Tracking software usage impacted purchasing
 - Client usage patterns not easily communicated
- **The Solution:** Capture, warehouse, and analyze
- **The Talent:** Early tenure project
- **The Outcome:**
 - Forecasting requirements for product acquisition and client usage is available.
 - CSV Importer

SMF Data Collection\Processing

- **The Problem:**
 - SMF data complexity across enterprise
 - Inconsonant resource usage
- **The Solution:** Analyze data collection
- **The Talent:** Early tenure project
- **The Outcome:**
 - Roadmap to standardization of SMF data
 - Leading to consistence experiences
 - Reduced processing time and CPU consumption



Mainframe Data Warehouse



- **The Problem:** Lack of a common warehouse repository to store sourced metrics in support of front office processes
- **The Solution:** Creation of the Mainframe Metrics Warehouse
- **The Talent:** Senior DBA aligned with me as the data architect.
- **The Outcome:** Single source of truth for mainframe related metrics being used throughout the organization.

Growing Pains

Culture Issues

- Not Invented Here
- Perceived Incompatibility

Connectivity Issues

- Simple TCP/IP Requirements
- Firewall / Security Engagement

Encoding Issues

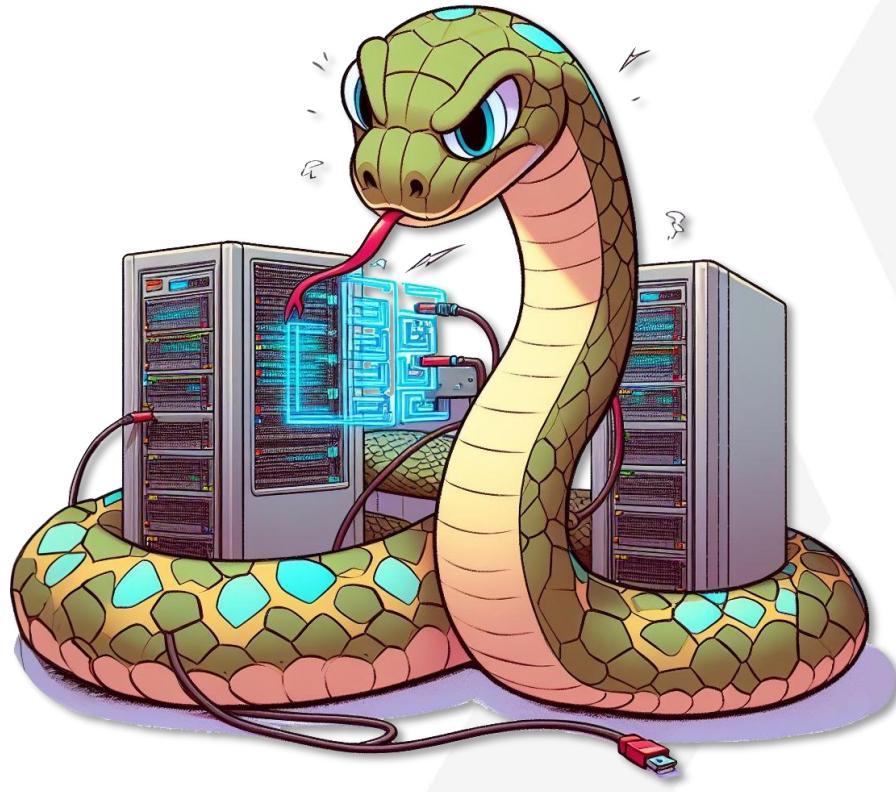
- File Tagging
- Read/Write Encoding (`encoding='cp1047'`)

Tech Debt Paydown

- Often Neglected USS Environments



Operationalizing Your Scripts



One Liner

```
//PYEX      JOB (01992),'Execute Python',
//          CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,REGION=0M
//SHELLCMD EXEC PGM=BPXBATCH,REGION=0M,
//          PARM='SH python3 /u/IBMUSER/myscript.py'
//STDOUT    DD SYSOUT=*
```

Multi-Line (Takes Arguments)

```
//PYEX      JOB (01992),'Execute Python',
//          CLASS=A,MSGCLASS=H,NOTIFY=&SYSUID,REGION=0M
//SHELLCMD EXEC PGM=BPXBATCH,REGION=0M
//STDOUT    DD SYSOUT=*
//STDPARM   DD *
SH python /u/IBMUSER/myscript.py
/*
```

What Does It All Mean?



Simpler, More Sustainable z/OS



Creating Technologists

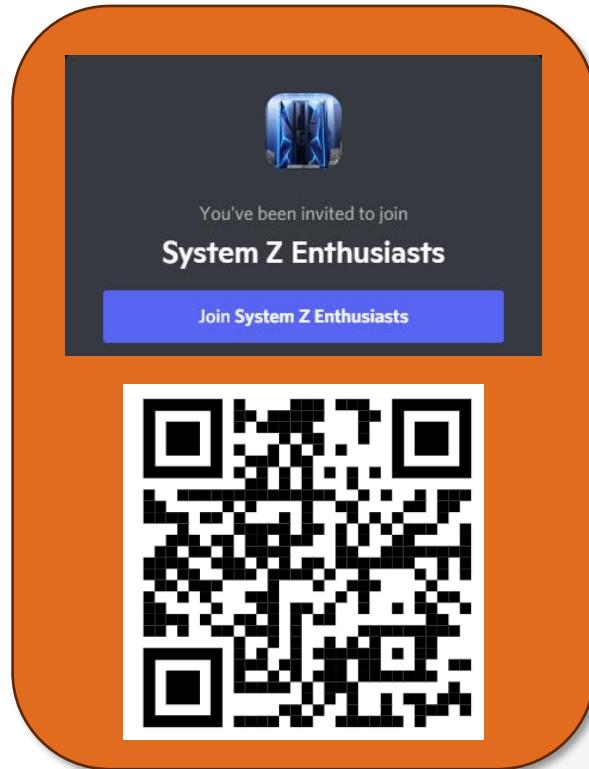


Achieving the Hybrid Cloud!



Before You Go

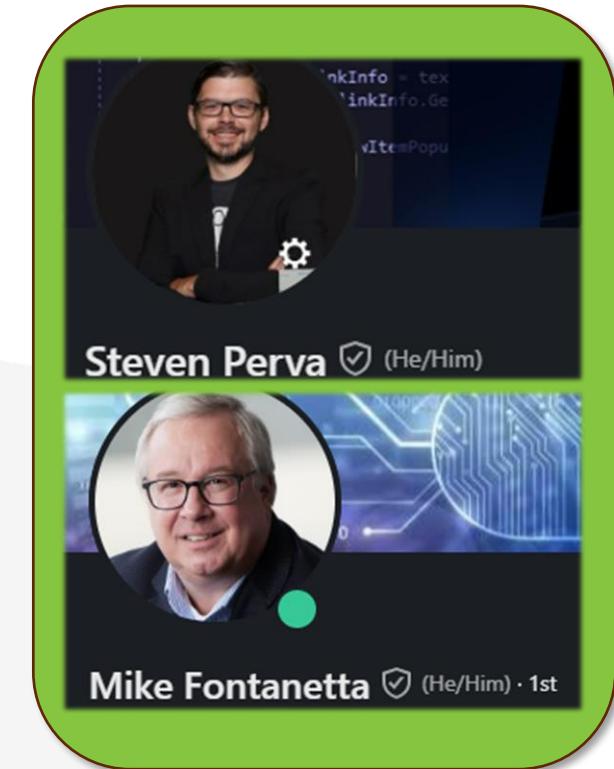
System Z Enthusiasts Discord



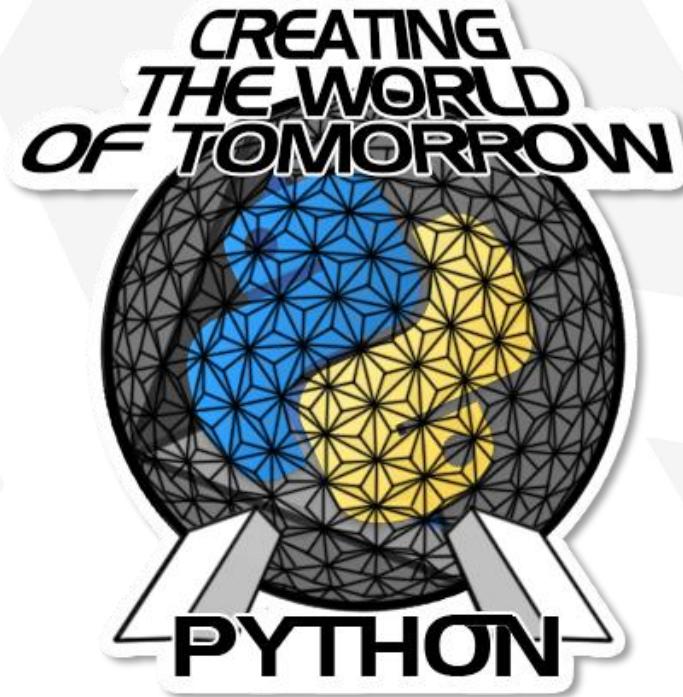
Python in z/OS Base IDEA



Connect on LinkedIn!



The End



I EF404 I SAY HELLO AND GET A STICKER

Your feedback is important!

Submit a session evaluation for each session you attend:

www.share.org/evaluation

SHARE mobile app

