zPET: IBM Z's First Customer

Michael Cohoon, Torin Reilly
June 10, 2019

IBM

Agenda

- Background on zPET
- zPET Environment
- Integration Test in Action -- Cloud Provisioning Example

What is zPET?

Z Platform Evaluation Test

- Z Platform Evaluation Test (zPET) is IBM Z's truest internal customer environment
- zPET runs a customer-like data sharing Parallel Sysplex on the IBM Z platform
- Focuses on reliability, availability, and serviceability (RAS)
- Runs on a mix of different machines and configurations
- Tests the integration of the full stack (hardware, firmware, software, middleware, applications, workloads, etc.)
- Expertise across numerous discipline with a team comprised of system programmers, operators, application developers, lab technicians, and more
- The unique makeup of the team allows us to take on projects that would be out of scope for other testing organizations

What is zPET’s goal?

- Mission: Verify that IBM Z is ready to support production, mission critical work as a platform
  - “The Final Verification”
- Ensures that the elements and features of z/OS work seamlessly together to support true production, mission-critical work
- From middleware expertise to automation, and from security setups to cryptography, zPET finds the integration problems before customers do
- Enterprise scale integration testing backed by high-volume, high-stress workload driven by more than 100 customer simulated and other applications running 24x7

zPET Software Stack

Proven tech…
- CICS TS and TG
- DB2 and IMS TM/DB
- WebSphere MQ, App Server
- WebSphere Process Server
- WebSphere Service Registry and Repository
- 25+ CICS, DB2, IMS and MQ Tools

Emerging tech…
- Open Data Analytics for z/OS
- z/OS Connect Enterprise Edition
- z/OS Cloud Provisioning and Management
- Common Data Provider for IBM Z
- IBM Machine Learning for z/OS

Monitoring…
- z/OS Management Facility
- Omegamon XE products… z/OS, CICS, IMS, DB2 PE, Messaging, Storage, MFN, JVM
- ITCAM for Transactions on z/OS
- DB2 Query Monitor for z/OS

Security…
- RACF
- zSecure Suite
- Multi-Factor Authentication for z/OS
- ISKLM, ICSF, Encryption Facility
- InfoSphere Guardium Data Encryption – DB2 and IMS
- Guardium S-TAP for DB2 and IMS

~100 products actively exploited and maintained in addition to BCP

Systems Mgmt…
- Tivoli Workload Scheduler
- System Automations
- Tivoli Netview for z/OS
- TDMF and zDMF
- Sterling Connect:Direct for z/OS

- GDPS PPRC, XRC and Active/Active*
- InfoSphere Q and SQL Data Replication

zPET Workloads

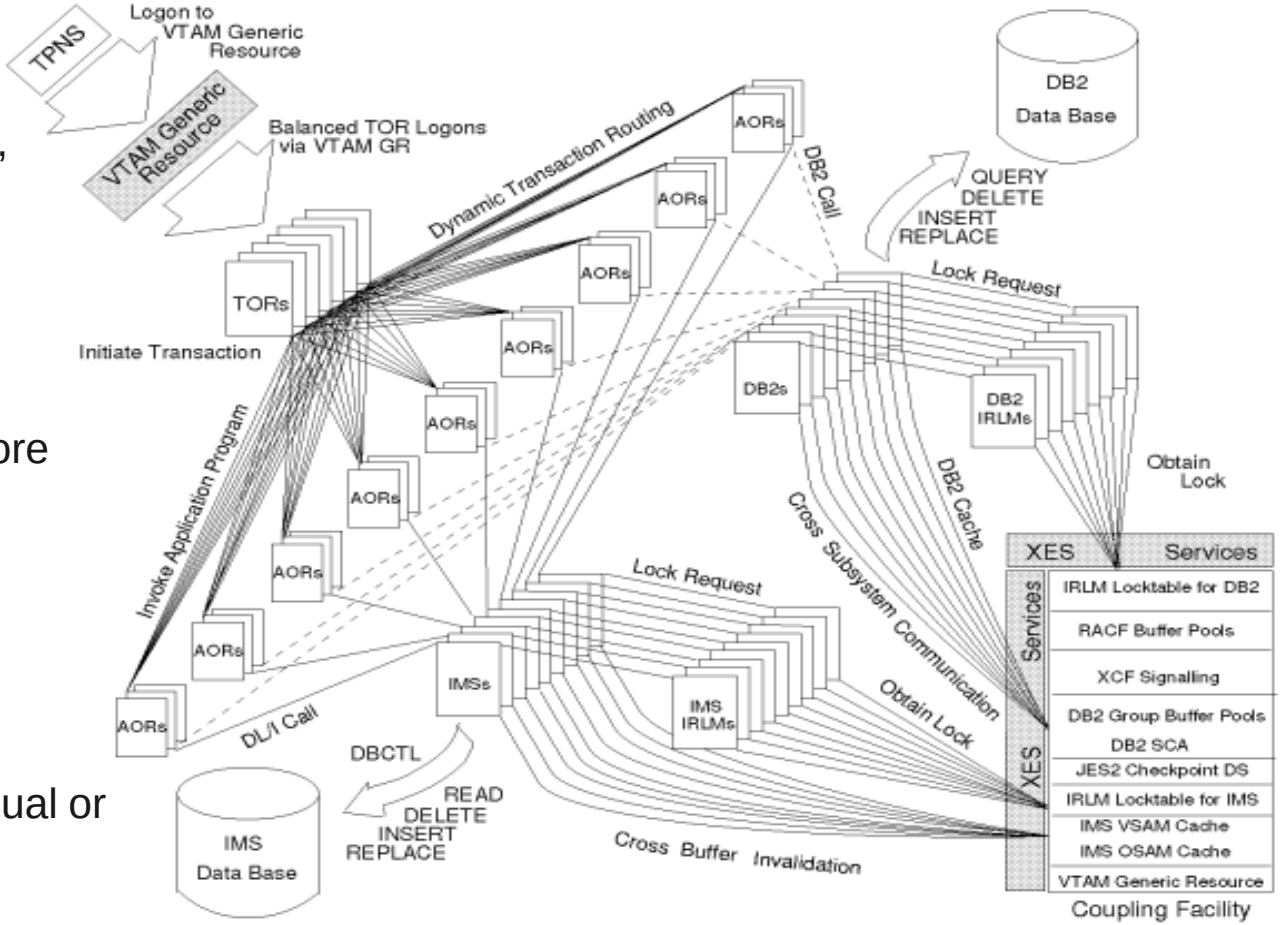
88+ workloads in various categories …

Customer Modeled: Simulates customer application flow, complex product integrations, highly available.

Component/Product Level: Focused on specific functionality.

Product Integration: Focused on integration of two or more products/components

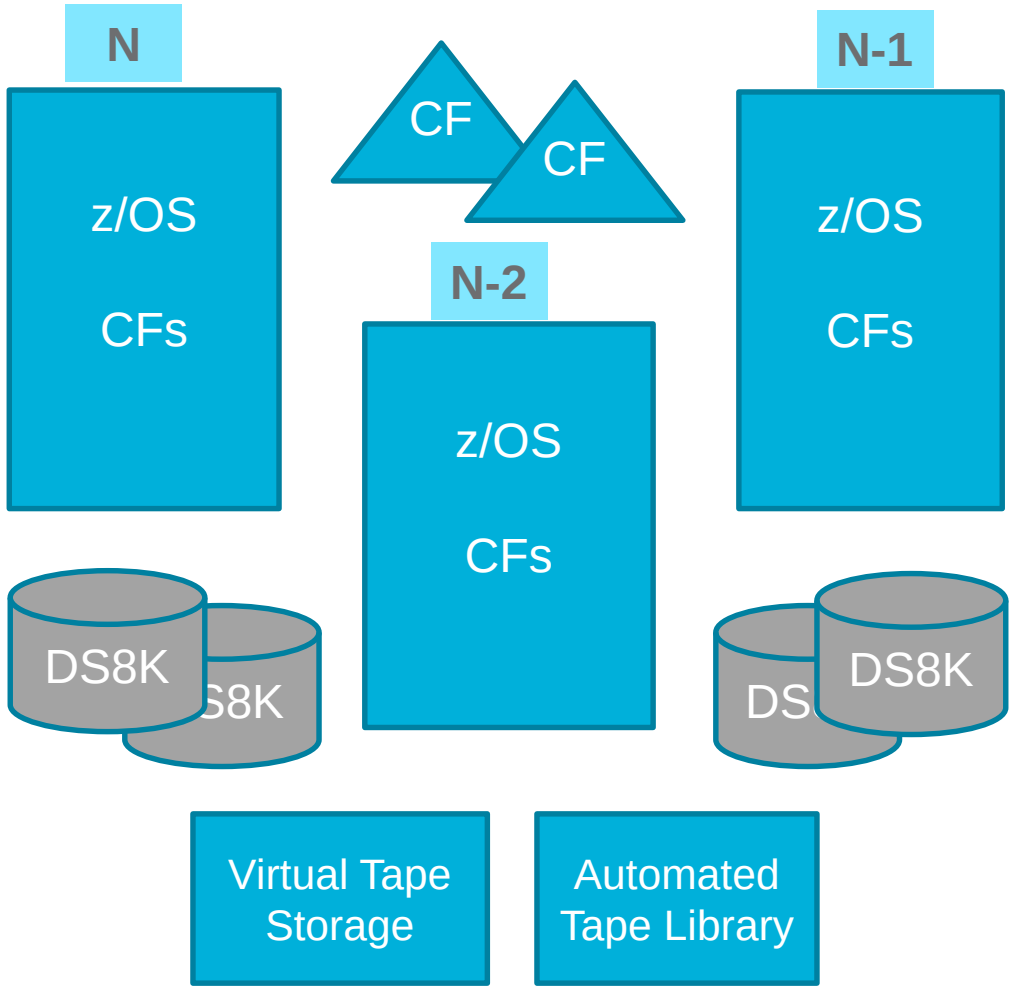
Crit-sit or APAR driven: Created to reproduce crit-sit problem or to validate APAR fixes. Run as regression.



For all these flows we do recovery tests by taking out individual or more components

- JES2, OMVS, TCP/IP, VTAM
- CICS TOR, AOR, CPSM
- IMS IRLM, DLI, DBRC, CQS
- DB2, IRLM
- SMSVSAM

zPET Infrastructure



The most robust Z environment within IBM!

Environment

- 16 & 4 way parallel sysplexes – **latest GA z/OS release**
- **N, N-1,** and **N-2** IBM z HW
- Mix of DS8Ks – **N, N-1,** and **others**
- **Automated Tape Library** and **Virtual Tape System**

Hardware features – Exploited as long as they are supported

- Crypto Express
- zHyperLink
- zEDC
- zHPF
- HyperSwap
- HiperSockets
- Flash
- RoCE
- HyperPAV
- XRC/SDM

Appropriately aggressive maintenance

- z/OS | Processor | Storage

Security Environment

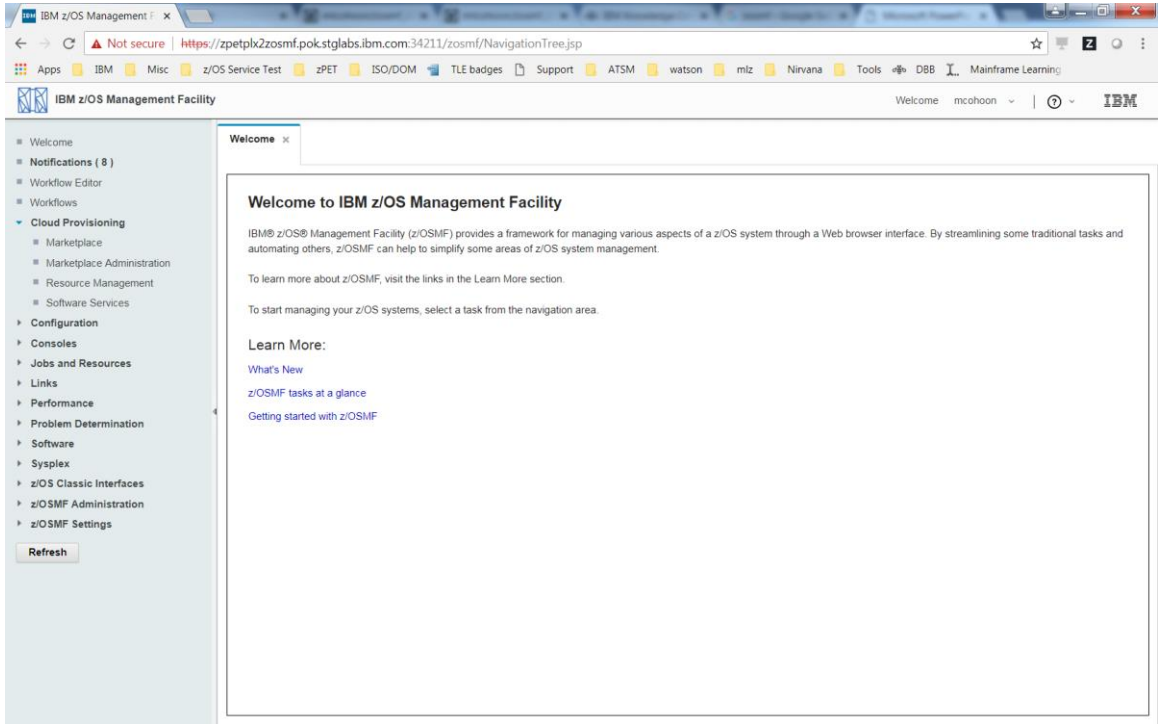
We run the following security products in our environment:

- RACF
- z/OS Integrated Cryptographic Service Facility (ICSF)
- IBM Tivoli Directory Server (LDAP)
- Encryption Facility for z/OS and OpenPGP
- Enterprise Key Management Foundation (EKMF)
- IBM Security Key Lifecycle Manager (ISKLM)
- PKI Services for z/OS

Cloud Provisioning

Background

- IBM z/OS Management facility (z/OSMF) delivers on IBM's strategy for mainframe simplification and modernization
- z/OSMF provides a modern browser based interface to managing the z/OS system
- The first release of z/OSMF was delivered as z/OSMF 1.11 at the same time as z/OS 1.11
- Starting with z/OS 2.2, z/OSMF ships with z/OS



"More than just a graphical user interface, the z/OS Management Facility is intelligent, addressing the needs of a diversified skilled workforce and maximizing their productivity."

"z/OSMF helps system programmers to more easily manage and administer a mainframe system by simplifying day to day operations and administration of a z/OS system."

IBM Cloud Provisioning and Management for z/OS

- Rapidly provision z/OS software subsystems
- Focuses on the ability to provision multiple workloads in a single z/OS instance
- Accomplished with core constructs:
  - Domains
  - Resource pools
  - Tenants
  - Templates
  - Instances
- Manage the provisioning of middleware resources that can be accompanied using REST APIs

This sounds great! But...

What will the impact be?

- Interfaces with various z/OS components and middleware
- Augment existing systems and processes, not replace them
- Tools for isolation
  - Storage, networking, procs
- Cannot affect existing workloads
  - Only add extra capability

Our value to the project

- Vast library of middleware used in production
  - DB2
  - MQ
  - CICS
  - IMS
  - z/OS Connect
  - WLP
- Experts across middleware and base products
  - Custom configuration
- Hardware
  - Crypto
  - etc.

zPET's CP&M Test Involvement

- Engage early in the release cycle
- Most customer-like environment due to:
  - Constantly exercised resources (CPU, storage, disk, etc.)
  - Many constant and current workloads running
  - Real environment instead of a development sandbox
- Execute provisioning of numerous different types of software instances (Db2, CICS, IMS, etc.)
- CP&M testing was added on top of a fully packed test schedule
  - Time consuming, manual work in addition to normal day jobs

Manual Testing Steps

1) Log in to z/OSMF (not a security admin user)
2) Create a new domain
3) Create a new tenant
4) Create a new resource pool
5) Create a new template
6) Approve the template
7) Switch users to the security admin and approve the template
8) Log back into the original user
9) Create a resource pool
10) Associate resource pool with the template
11) Create a network resource pool
12) Create a port range
13) “Complete” the network resource pool
14) “Complete” the resource pool
15) Publish the template
16) Run the template
17) Ensure the instance was properly provisioned
18) Deprovision the instance
19) Remove all created resources

How can we make this simpler?

Shell scripting

- z/OSMF defines REST APIs to interact with services
- Began with simple GETs using cURL
- Problems:
  - Single UI actions complex behind the scenes
  - Massive response bodies
  - Timestamp conversion
  - State management
  - z/OSMF plugins APIs can differ

REST interface	Identifier
Application Linking Manager interface	APPLINK
Application server routing services	GATEWAY
Cloud provisioning services	PROVISIONING
Data persistence services	PERSIST
Multisystem routing services	GATEWAY
Software management services	SWMGMT
Topology services	SYSTEM
TSO/E address space services	TSO
z/OS data set and file REST interface	FILES
z/OS jobs REST interface	JOBS
z/OSMF workflow services	WORKFLOW

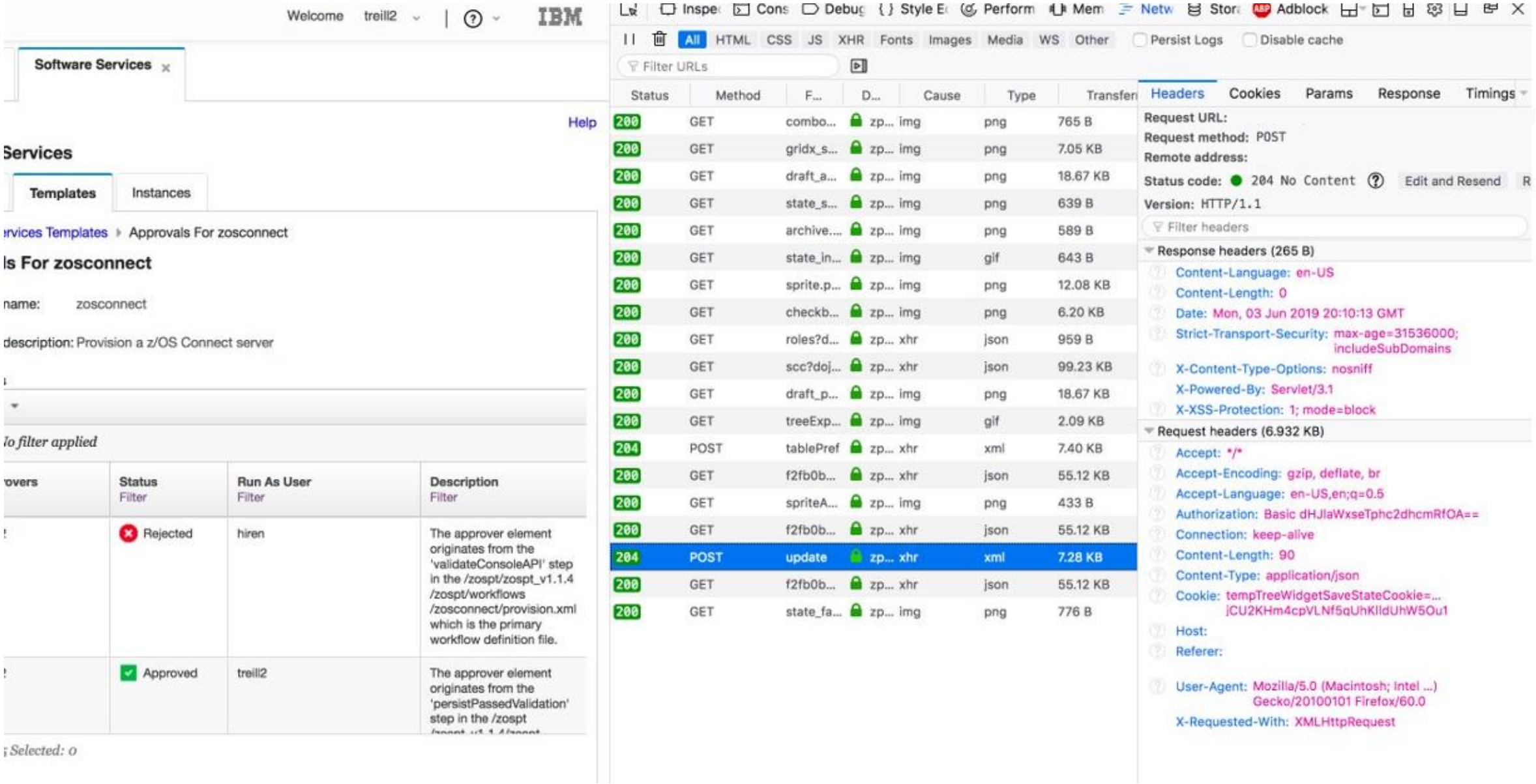
Python

- Chained requests together in Python (ie: create, view, and delete a resource)
- Maintain state through objects
- Run numerous threads of execution in parallel
  - Simulate concurrent users (while resources are exercised)
- Robust testing framework
- Allows for simple provision properties file
- Translated manual testing process into API requests

```
[VARIABLES]
WLP: WebSphere_Liberty_Profile_template.config
zCEE: zOS_Connect_template.config
QMgr: MQ_Queue_Manager_template.config
MQ: MQ_Queue_template.config
CICS: CICS_54_template.config


[PLAN]
WLP
zCEE
QMgr
MQ
CICS
WLP, zCEE, QMgr, MQ, CICS
WLP, zCEE, QMgr, MQ, CICS
WLP, zCEE, QMgr, MQ, CICS
WLP, zCEE, QMgr, MQ, CICS
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
```

Biggest hurdle – undocumented APIs!!



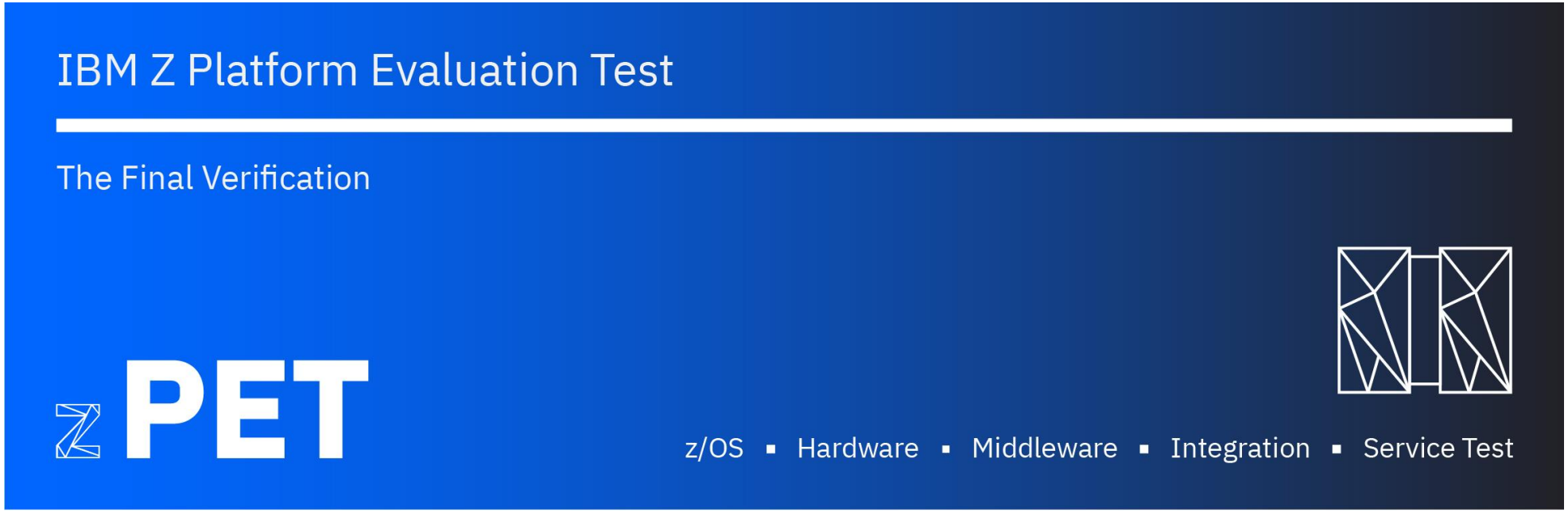
Value created

- More test coverage
  - Daily regression
- "Smoke test" for new versions
- Introduces randomness
  - Concurrent provisions conflicts
  - Timing defects
- Feedback to development to drive improvements
  - Externalize more APIs

zPET – IBM Z Platform Evaluation Test

- Integration test the IBM Z platform in a customer like environment
  - HW, OS, SW stack, workloads and processes
- Act as z/OS and IBM Z HW deliverables' first customer; a sponsor user
- Continuous improvement culture generates tremendous implicit test value
  - Integrate new functions, features, products into the existing environment, adjust workloads to exploit them, just like our clients will
  - Even without explicit test execution – integrating and exposing all latest hardware, z/OS, ~100 SW products, functions and features, as well as the latest maintenance to one another … 24x7x365
- Explicit testing focused on role based end user scenarios… typically migration, setup, and recovery heavy
- Lots of additional sharing of the environment and data
  - Live demos
  - Environment used for collateral
  - Operational data provided to dev for analytics & POCs
- Provide around the sun operational and execution coverage

zPET Blog



Visit our Blog at: https://ibm.biz/zPETBlog

- Articles
- Tricks, tips, and hints
- User experiences
- REXX execs
- JCL
- Parmlib members

Questions?