Deploying CICS regions with the z/OS Provisioning Toolkit

Dan Millwood - https://www.linkedin.com/in/dan-millwood-32373042/
IBM UK Ltd

November 2018
Session GL

Important Disclaimer

IBM's statements regarding its plans, directions and intent are subject to change or withdrawal without notice at IBM's sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

•THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided. It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs.

•**Performance**. Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

•**Customer Examples**. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

•**Availability**. Not all offerings are available in every country in which IBM operates. This document is current as of the initial date of publication and may be changed by IBM at any time.

•**Trademarks.** IBM and the IBM logo are trademarks of International Business Machines Corporation, registered in many jurisdictions. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Other company, product and service names may be trademarks, registered marks or service marks of their respective owners.

I have an idea!

Drawing by TomBrough openclipart.org

z/OS Provisioning Toolkit

A modern solution for the rapid provisioning of z/OS development environments.

Application developers can provision and deprovision z/OS development environments, customized for their applications, in minutes, without requiring any z/OS specific administration skills.

System programmers maintain control of the provisioning process by creating templates for provisioning subsystems using z/OSMF workflows

Self service provisioning can be enabled through use of the **IBM Cloud Provisioning and Management for z/OS** plug-in for the **IBM z/OS Management Facility** (z/OSMF), which controls developer access and sets appropriate provisioning limits.

The toolkit is fully supported and available now to all z/OS V2 clients at **no additional charge**.

z/OS Provisioning Toolkit

What's in the box?

Automation workflows:
- Provision / deprovision CICS
- Provision / deprovision z/OS Connect EE

The workflows run in a workflow engine provided by z/OSMF

Command line tool **zospt**
- **Build image** describing subsystem configuration
- **Run** image to provision configured subsystem

zospt builds images describing how subsystems should be configured.
zospt drives z/OSMF through its REST API to provision subsystems and customize their configuration

Additional workflows compatible with **zospt** are available for IBM MQ and WebSphere Liberty

IBM supplied CICS workflows

How can they help?

- Repeatable template based provisioning
- Reduce risk of errors
- Simplify configuration
- Fast adoption of new features
- Provision and deprovision
- New CICS regions within minutes
- Integration with z/OSMF self-service framework
- Reduce maintenance costs
- Follow best practice

What can the CICS workflows do?

An introduction to the capabilities in the automation

Provision

- CICS TS V5.1+
- CMAS and CICSplex
- MAS / WUI into existing CICSplex
- Individual regions with / without CICS Explorer connectivity

Customize

- The provisioning process
- The configuration of the CICS regions
- Whether to use a pre-existing CSD, or create a new one for each CICS region

Connect

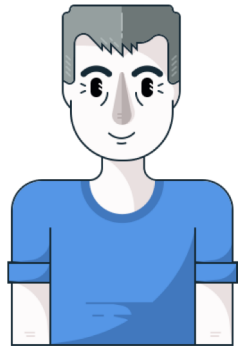
- CICS to existing DB2
- CICS to existing MQ Qmgr
- CICS to provisioned MQ Qmgr
- z/OS Connect EE to provisioned CICS region

Capabilities

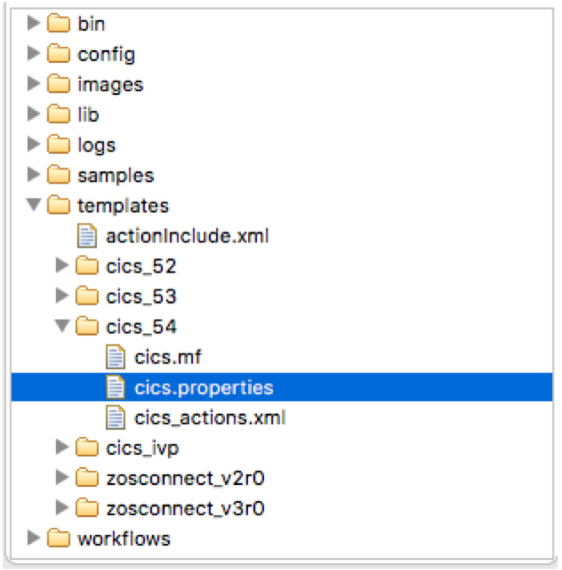
- Add embedded Liberty JVM server listening on http / https / debug port
- Add support for SOAP based Web services
- Add support for JSON based Web services
- Add support for Node JS applications

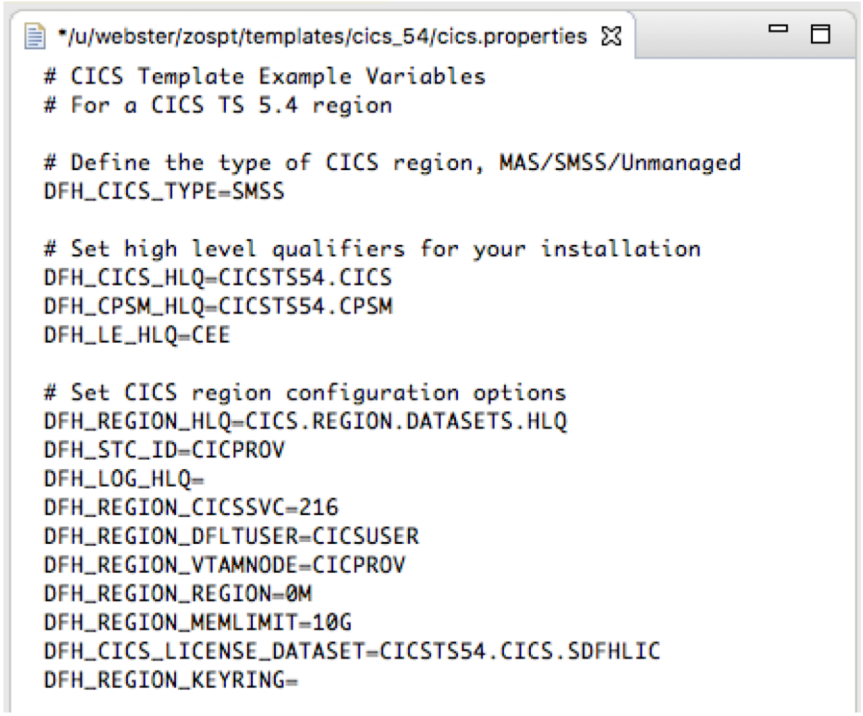
The CICS Workflows

How are they configured?



Stan Cicero
Senior System Programmer





Stan fills out a simple properties file to describe his environment.

Evaluating the workflows

Use the getting started scenario in z/OS PT 1.1.2

Systems Programmer

- Provision a CICS region
- Evaluate workflows
- Evaluate zospt
- Minimal pre-reqs

Systems Programmer

- Create a simple CICS template
- Provision a CICS region
- Evaluate z/OSMF cloud provisioning

Application Developer

- Self-service provision CICS regions
- Full function z/OSMF cloud provisioning

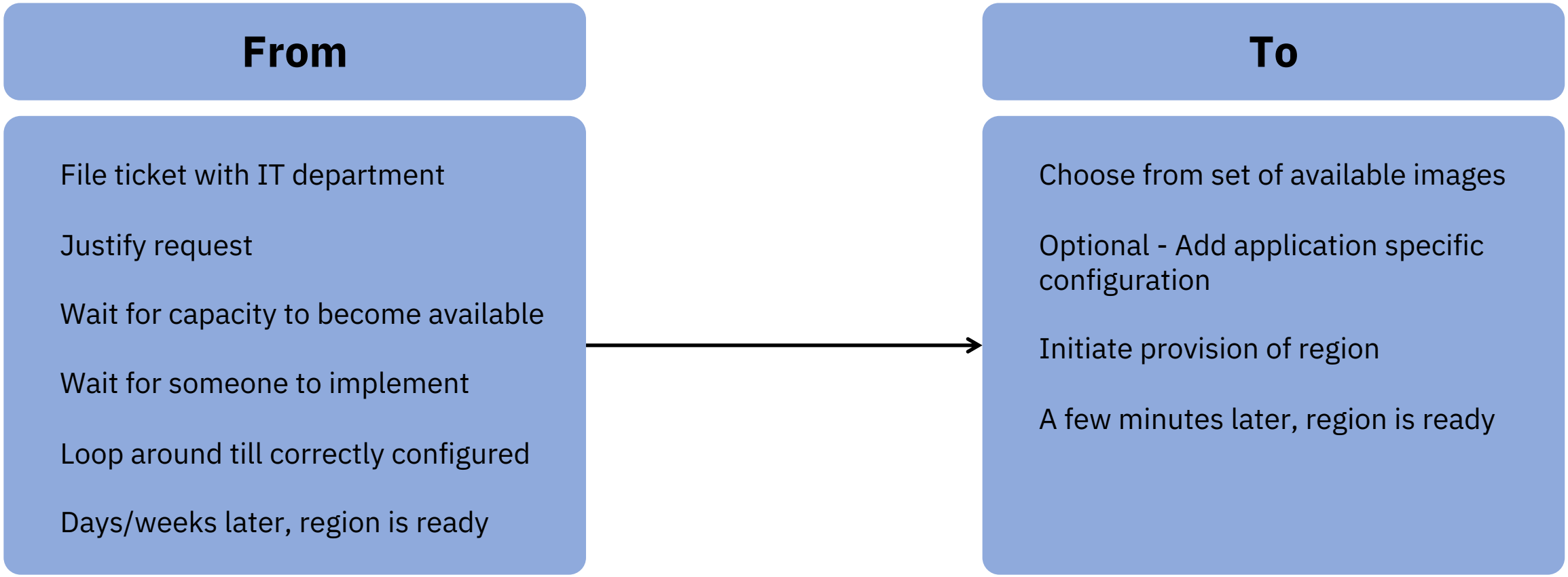
Application Developer

- Self-service provision different CICS configurations
- zospt images + z/OSMF cloud provisioning

Evaluate workflows and zospt → Evaluate z/OSMF Cloud provisioning → Create a self-service CICS template → Provision customized regions using zospt images

Self-service provisioning

Accelerating development and test



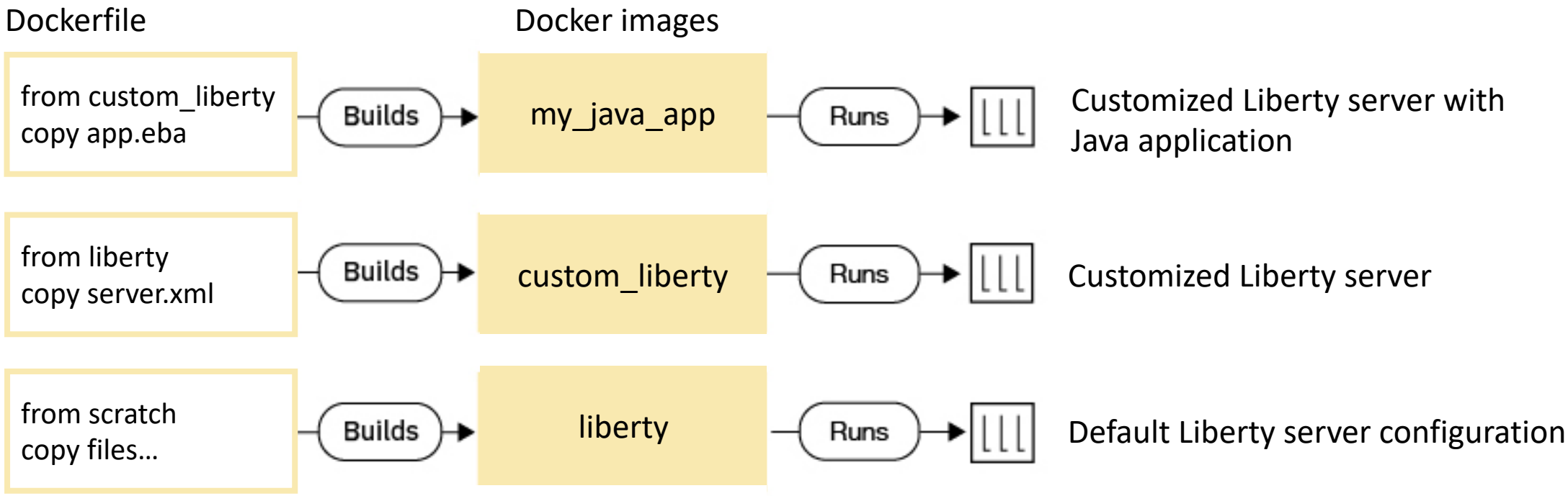
Self-service provisioning

What does it really mean?

- The provisioning process is fully **automated**.
  - Subsystems are available on demand and can be deprovisioned when no longer required.
- **Approvals** for new subsystems are agreed up front.
  - *"We agree that over the next year product team X can use a maximum of 10 regions."*
- System programmers still maintain **control** over the configuration of the regions.
  - Limited freedom to customize the regions for specific applications could be granted.
- The automation that is run during provision and deprovision is **reviewed and approved**.
  - The automation steps are run under agreed automation ids.
- **Network resource pools** are available for the provisioned regions.
  - Regions can be assigned ports and APPLIDs from a pool.

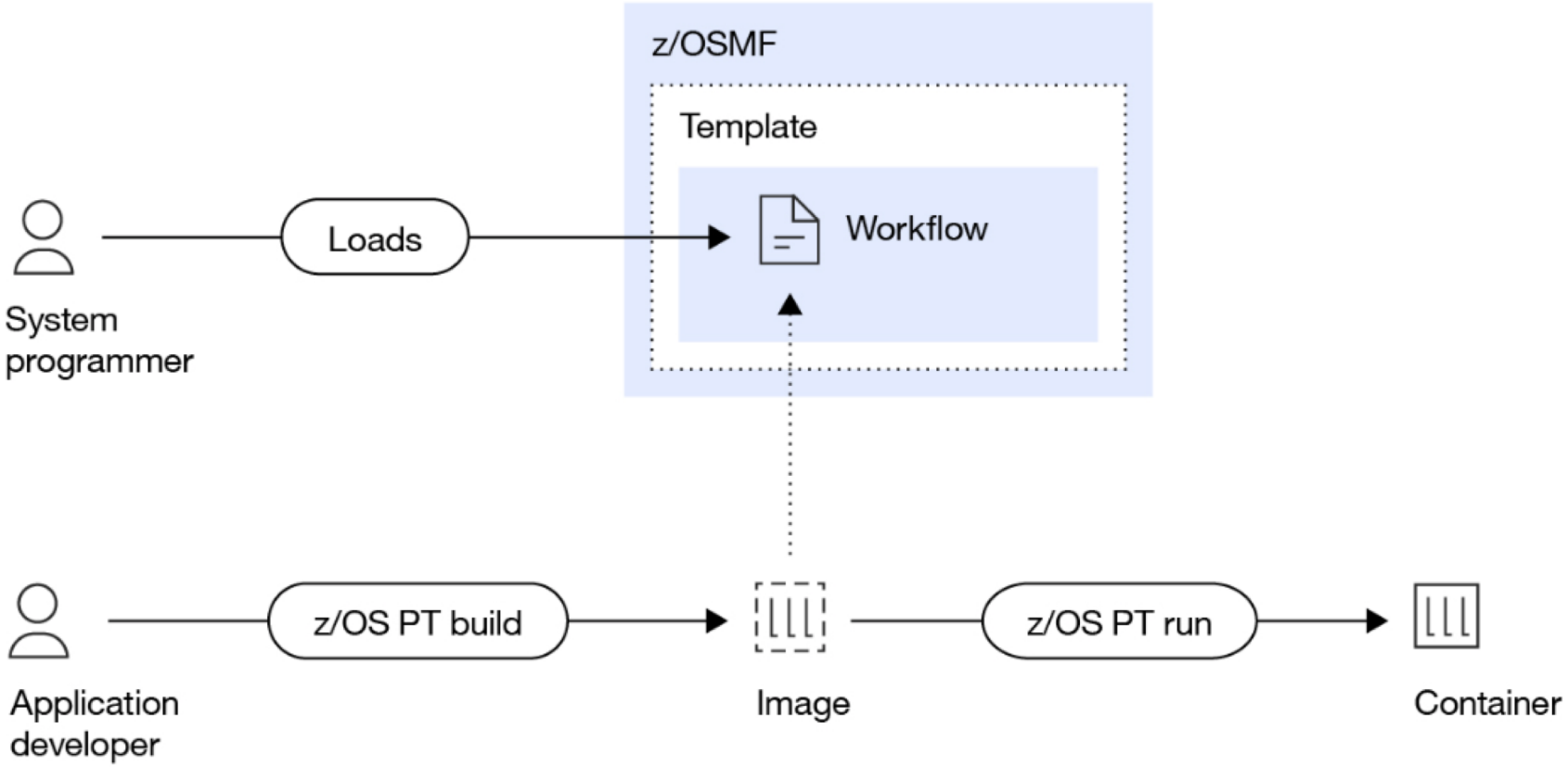
Self-service provisioning

How is this achieved on other platforms?



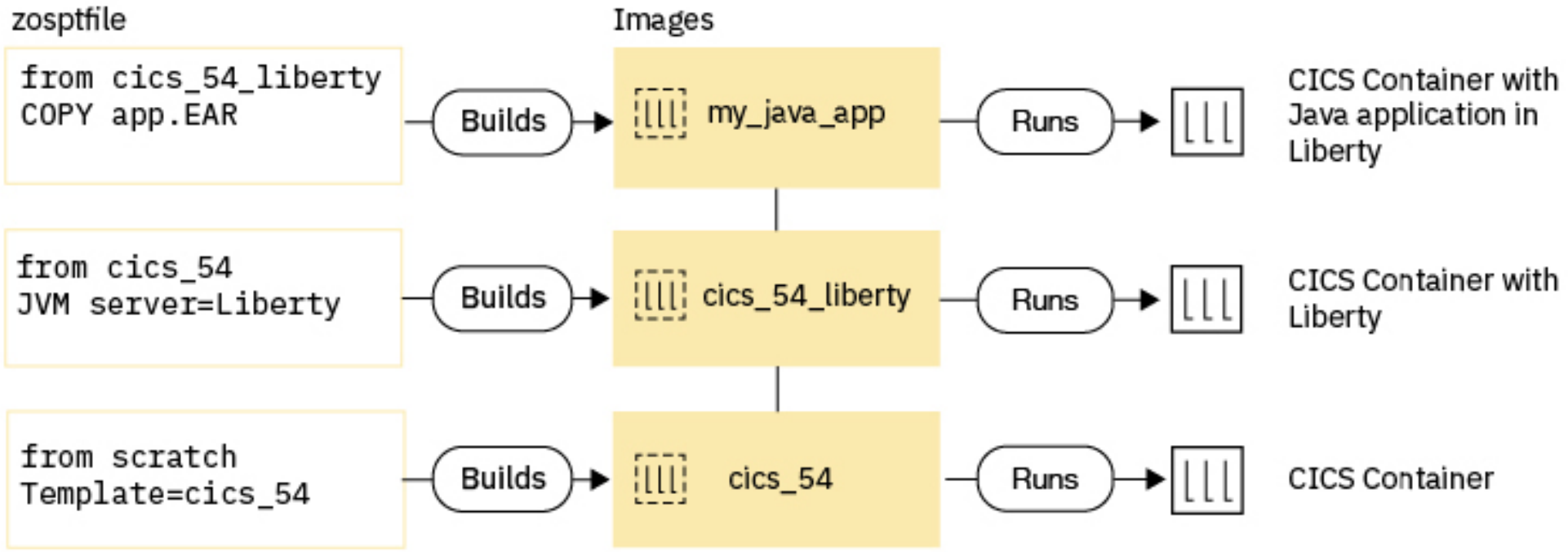
Self-service provisioning

How the z/OS Provisioning Toolkit works



Self-service provisioning

Using images with the z/OS Provisioning Toolkit



Use Case – Using a pre-defined image

Alan provisions a region running the insurance app using an image provided by a system programmer

Alan Andersson

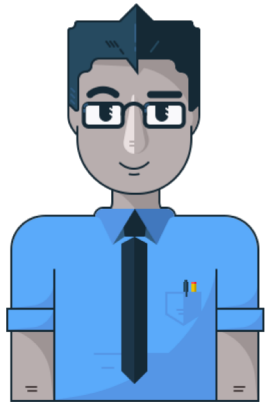
COBOL App Developer

```
/u/alan:>zospt images
2018-04-12 12:01:20  IBM z/OS Provisioning Toolkit V1.1.0
IMAGE NAME                CREATED                SIZE
cics_bank_app             2018-04-11T14:30:41  10240B
cics_insurance_app        2018-04-11T14:30:41  10240B
cics_no_app               2018-04-11T14:30:41  10240B
```

```
/u/alan:>zospt run cics_insurance_app --name alan
2018-04-12 12:06:36  IBM z/OS Provisioning Toolkit V1.1.1
2018-04-12 12:06:36  Running image cics_insurance_app.
...
2018-04-12 12:07:55  Container alan has been started.
```

Use Case - Building custom images

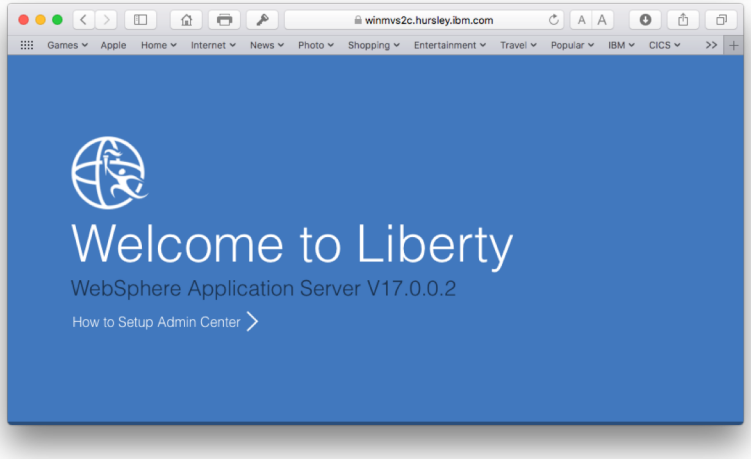
Jash wants to provision CICS + Liberty + his application



Jash Jayaraman
Java App Developer

```
FROM cics_54_liberty
COPY app.bundle_1.0.1 bundles/app.bundle_1.0.1
```

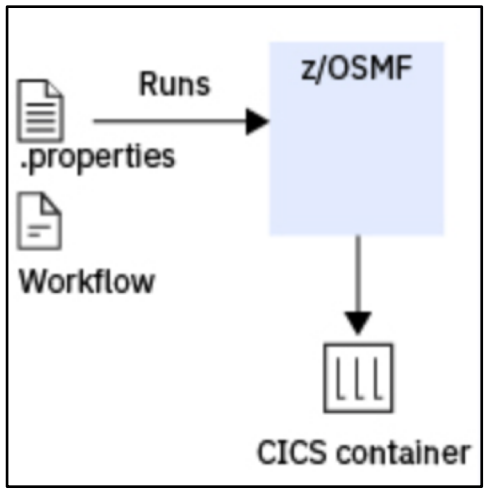
```
bin — ssh millwoo@winmvs29.hursley.ibm.com — 188×20

/u/millwoo/provisioningQ4_17/myLibertyApp:>zospt build -t jash_app .
2018-04-12 11:52:59  Building /u/millwoo/provisioningQ4_17/myLibertyApp/zosptfile into image jash_app.
2018-04-12 11:52:59  Step 0 : FROM cics_54_liberty
2018-04-12 11:52:59  Step 1 : COPY app.bundle_1.0.1 bundles/app.bundle_1.0.1
2018-04-12 11:52:59  Directory name=app.bundle_1.0.1, path=rootfs/bundles/app.bundle_1.0.1.
2018-04-12 11:52:59  Directory name=META-INF, path=rootfs/bundles/app.bundle_1.0.1/META-INF.
2018-04-12 11:52:59  File name=cics.xml, path=rootfs/bundles/app.bundle_1.0.1/META-INF/cics.xml.
2018-04-12 11:52:59  File name=.cics.xml~, path=rootfs/bundles/app.bundle_1.0.1/META-INF/.cics.xml~.
2018-04-12 11:52:59  File name=com.ibm.cics.server.examples.wlp.tsq.ebabundle, path=rootfs/bundles/app.bundle_1.0.1/com.ibm.cics.server.examples.wlp.tsq.ebabundle.
2018-04-12 11:52:59  File name=com.ibm.cics.server.examples.wlp.tsq.app.eba, path=rootfs/bundles/app.bundle_1.0.1/com.ibm.cics.server.examples.wlp.tsq.app.eba.
2018-04-12 11:52:59  File name=DAN1.transaction, path=rootfs/bundles/app.bundle_1.0.1/DAN1.transaction.
2018-04-12 11:52:59  File name=WEBAPP.urimap, path=rootfs/bundles/app.bundle_1.0.1/WEBAPP.urimap.
2018-04-12 11:52:59  File name=.DAN1.transaction~, path=rootfs/bundles/app.bundle_1.0.1/.DAN1.transaction~.
2018-04-12 11:52:59  File name=.WEBAPP.urimap~, path=rootfs/bundles/app.bundle_1.0.1/.WEBAPP.urimap~.
2018-04-12 11:52:59  File name=.com.ibm.cics.server.examples.wlp.tsq.ebabundle~, path=rootfs/bundles/app.bundle_1.0.1/.com.ibm.cics.server.examples.wlp.tsq.ebabundle~.
2018-04-12 11:52:59  File name=.com.ibm.cics.server.examples.wlp.tsq.app.eba~, path=rootfs/bundles/app.bundle_1.0.1/.com.ibm.cics.server.examples.wlp.tsq.app.eba~.
2018-04-12 11:52:59  Successfully built /u/millwoo/provisioningQ4_17/myLibertyApp/zosptfile into image jash_app.
/u/millwoo/provisioningQ4_17/myLibertyApp:>
```



Jash builds his application using a simple zosptfile and can provision his environment in 3 minutes…

Take that first step

Trying the CICS getting started scenario



- Use the zospt command line utility to run a workflow in z/OSMF and provision a CICS region
- Pre-requisites
  - Access granted to run workflows in z/OSMF in a sandbox environment
  - Authority in the sandbox to:
    - Allocate data sets.
    - Add a procedure into a PROCLIB.
    - Issue console commands.
  - An APPLID that can be used by the provisioned CICS region

Take that first step

Trying the CICS getting started scenario

1. Download and unpax the zospt command line tool and workflows
2. Configure the connection from zospt to z/OSMF
3. Fill in the cics.properties file to configure the provisioning process
4. Run the provision using the zospt command line tool

- Each step in the provisioning process submits a JES job
- Each step in the provisioning process runs under the users own ID
- If a step fails, you can use the submitted JCL and the job output to help diagnose why

Demo

Trying the CICS getting started scenario

Resources to help you get started

Provisioning CICS made simple!

- Download z/OS PT from
  - z/OS PT product page
- Blog posts
  - A step by step guide to provisioning your first CICS region from a 3270 terminal
  - Customizing your z/OS PT provisioned CICS regions
- Videos
  - https://www.youtube.com/watch?v=PyznEBFajKg
  - https://www.youtube.com/watch?v=_q48arq828Y
- Knowledge Center
  - Getting started with CICS by using z/OSMF workflows
- IBM Cloud Provisioning and Management for z/OS
  - Enhancements delivered in June 2018 to improve the getting started experience
  - Product Page

We want your feedback!

- Please submit your feedback online at ….
  - ➢http://conferences.gse.org.uk/2018/feedback/GL

- Paper feedback forms are also available from the Chair person

- This session is GL