LDUG

26th September 2024

Rob Gould – rob.gould@triton.co.uk

Agenda

- OCP on Z
- Ansible
- Git
- VSAM Migration

OpenShift Container Platform

Agenda

- What is it?
- Cool – but why do I care?

OCP – What is it?

Once upon a time, in a galaxy far, far away… (the 1960s)

- Computers ran a single program from start to finish
- Batch focused
- Not a great use of all of that horsepower, so…

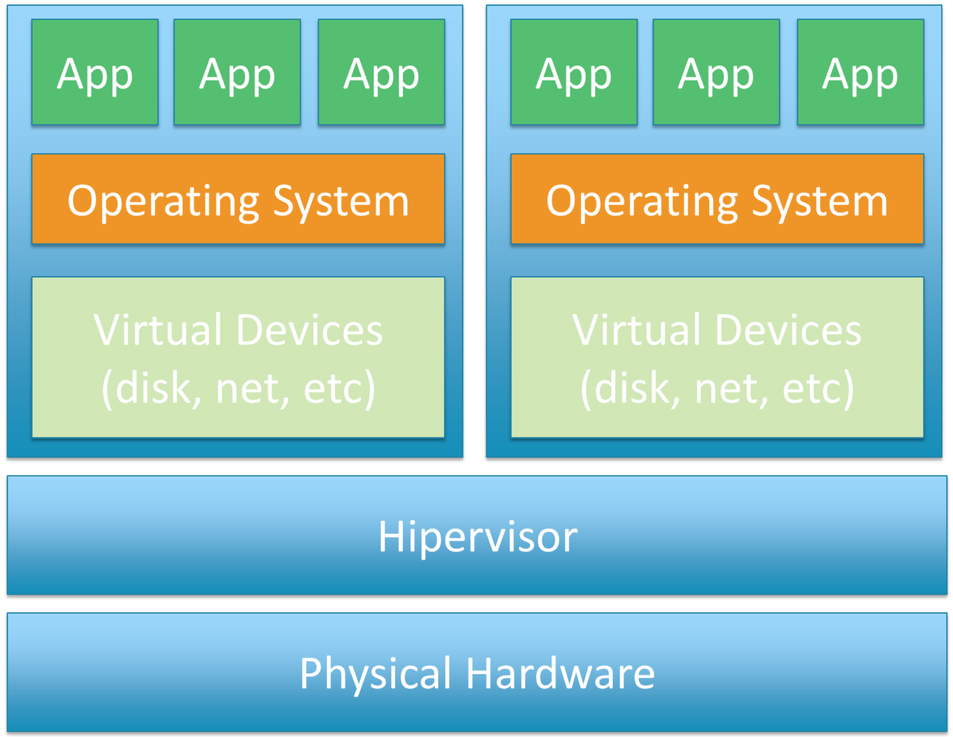
Time sharing

- More than one program running at a time
- Shared resources
- Dispatcher decides which programs get the processor
- But, as machine capacity increased, lots of idle horsepower, so…

OCP – What is it? (2)

Virtualisation

- Hardware / physical abstraction
- Hipervisor controls the distribution of physical resources to logical / virtual machines
- E.g.
  - PR/SM
  - VM
  - VMWare
  - Virtual Box



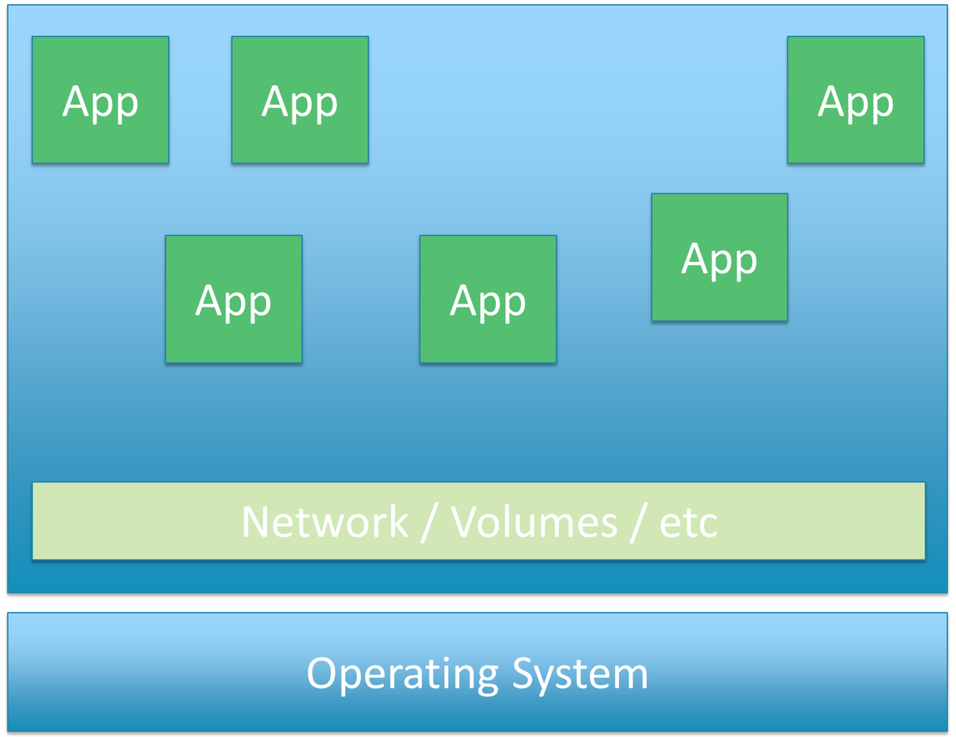
OCP – What is it? (3)

Containerisation

- Operating system abstraction
- Applications run in separate memory contexts (containers)
- Shared virtual network
- Mountable host paths (volumes)
- Examples:
  - Docker (root requirements)
  - Podman (user land)

So what?

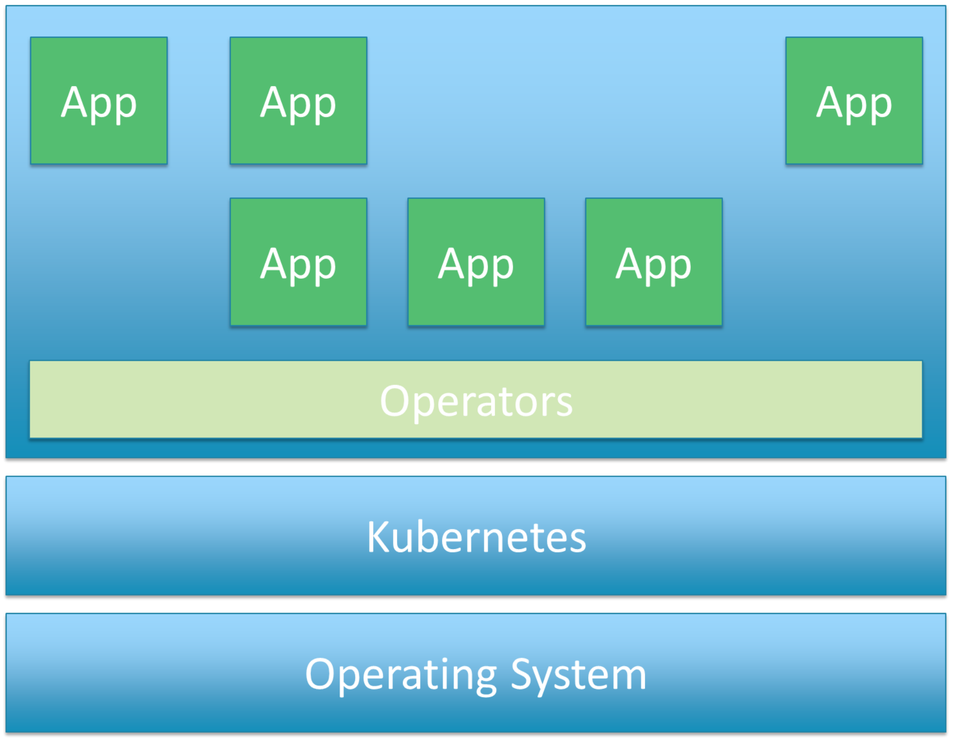
- Fully portable functional applications
- Dependencies packaged with app
  - E.g. libstdc++, libcrypt, etc
  - = app stability



OCP – What is it? (4)

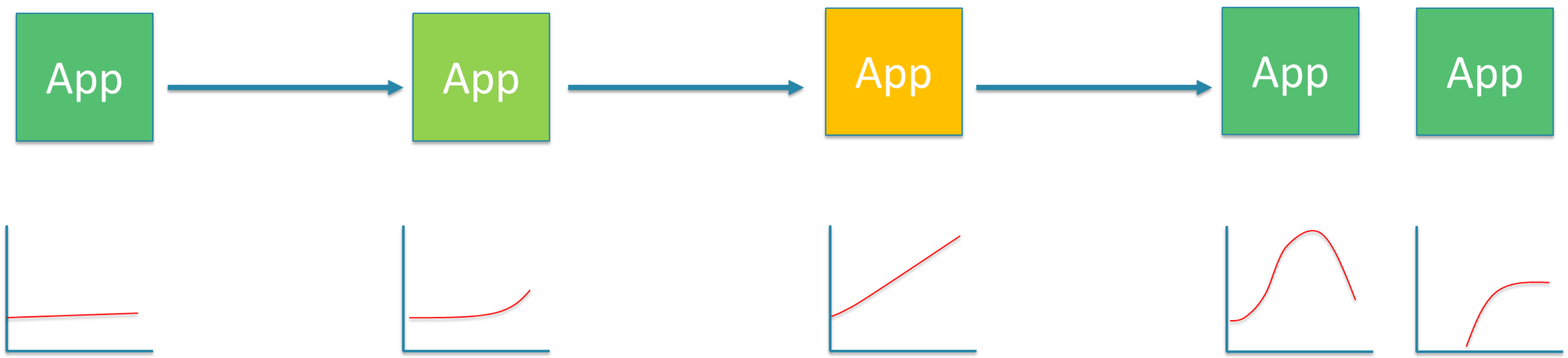
OpenShift

- Kubernetes managed cluster of application and operator containers
- Integrated cluster services
  - Monitoring
  - Role based access
  - Event trap / trigger
  - Automation (Tekton)
  - Logging
  - Etc
- Application services
  - Service mesh
  - Cloud Paks
  - Git Hub integration
  - Argo CD
  - etc



OCP – What is it? (5)

- Scalability
  - Horizontal (load driven ramp up the number of pods running within the node)
    - Demand driven
      - CPU consumption
      - Memory consumption
    - Automation / external triggers
  - Vertical (more nodes)
    - Working on it (not out of the box)



OCP – Cool, but why do I care?

Available on Z (LoZ under KVM or z/VM)

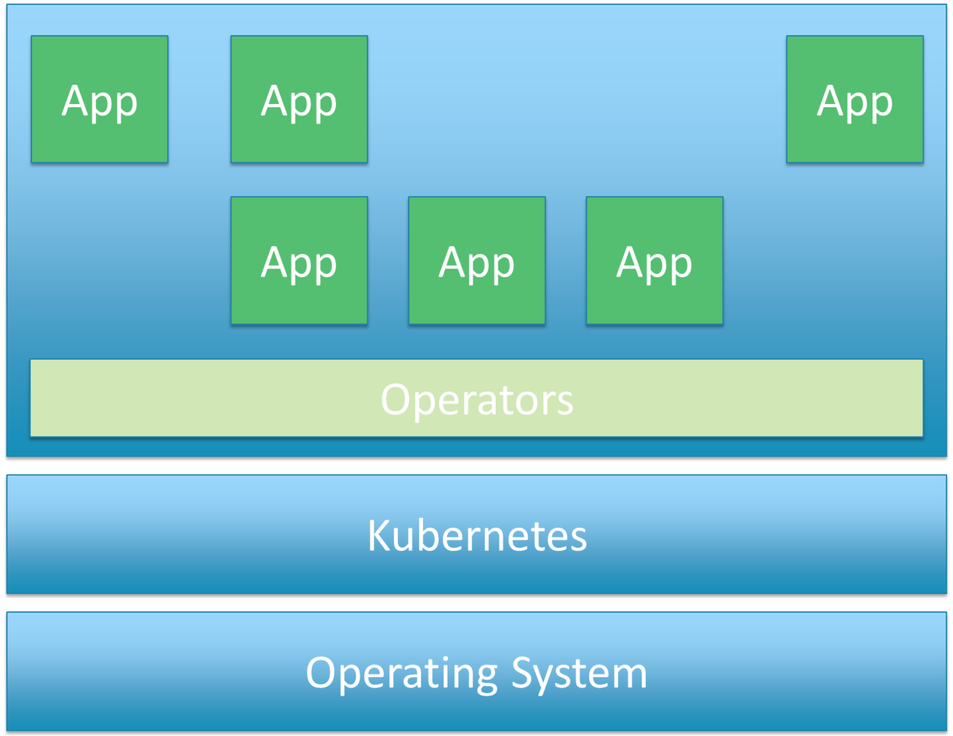
- Data gravity – connection to Db2 for z/OS over Hipersockets (microseconds not milliseconds)
- Platform quality (99.9999% uptime, DR already in play, surrounding infrastructure)

Available on major Cloud providers

- AWS
- MS Azure
- Google Cloud
- IBM Cloud

CP4d

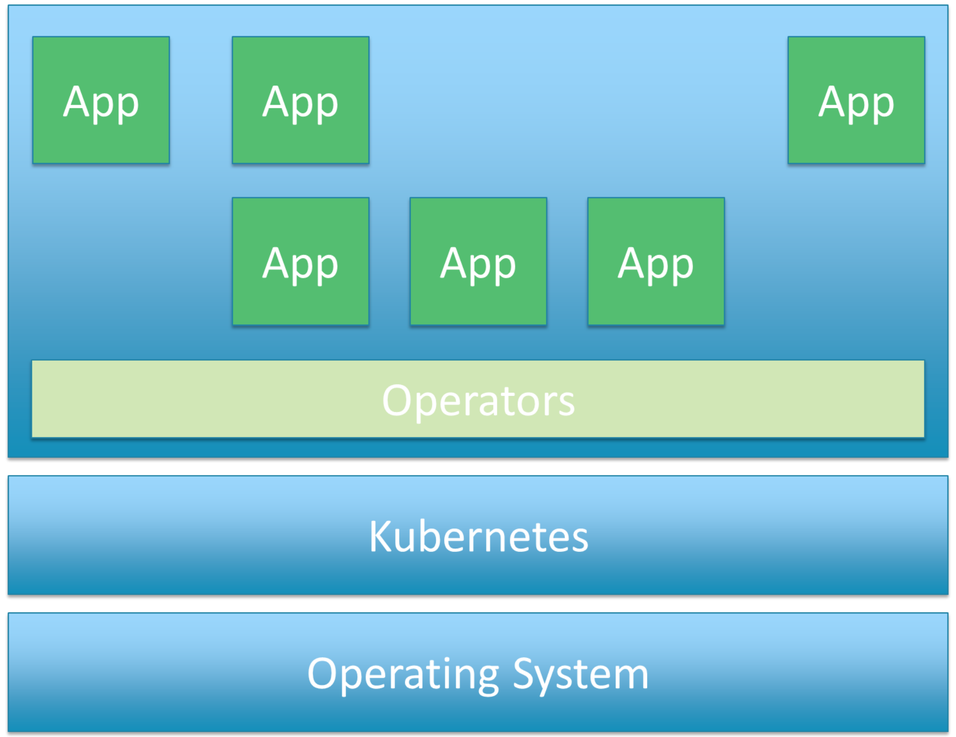
- Data Gate
- Db2 LUW
- etc



OCP – Cool, but why do I care? (2)

Automation:

- E.g. tag based deployment:
  - Create a new version of your app – app1:1.0.1
  - Deliver the container image (image stream) to OCP
  - Move the “latest” tag to 1.0.1
  - Kubernetes will quiesce and restart pods
  - Different strategies can be configured
- Tekton pipelines
  - Build apps
  - Manage operation
  - Triggered by events
    - E.g. internal (Prometheus)
    - Or external (GitHub)



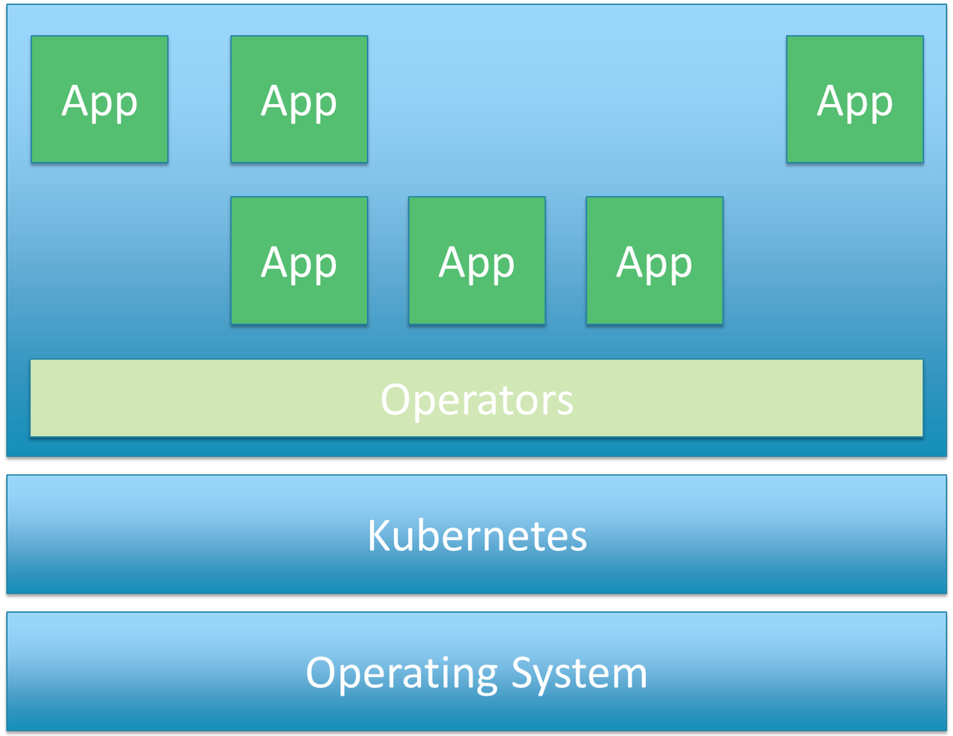
OCP – Cool, but why do I care? (3)

Configuration Maps

- Data shared with the app container – mapped as a file path
- E.g. connected Db2 service host name and port

Secrets

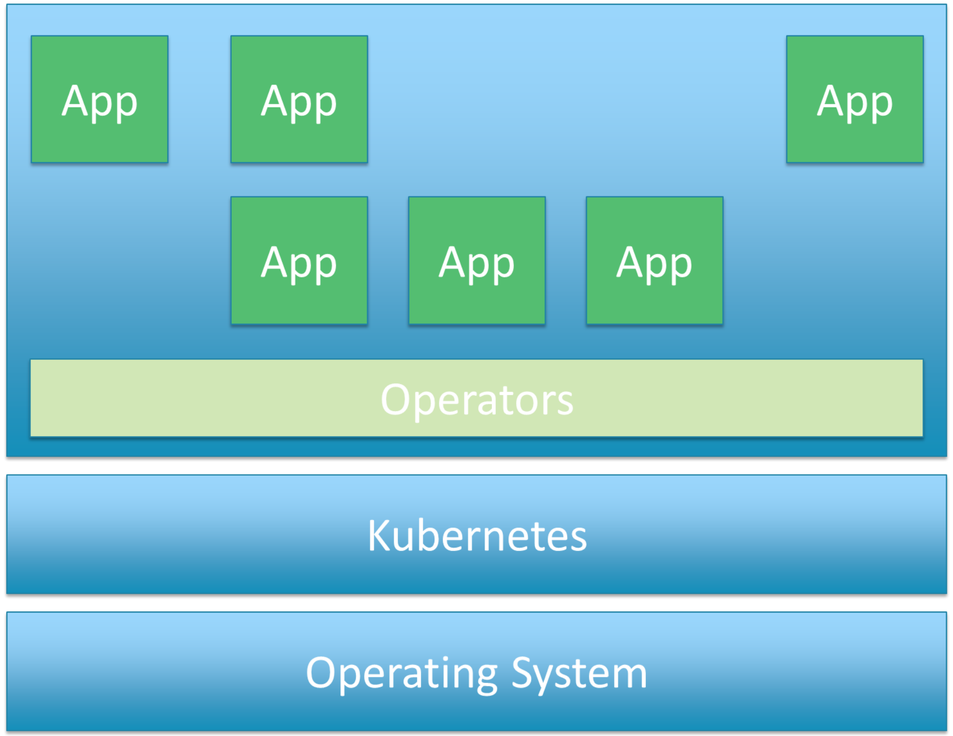
- Like ConfigMaps, but only visible to app
- E.g. Db2 service connection certificate



OCP – Cool, but why do I care? (4)

Operators:

- Extending functionality inside OCP – e.g.
  - GitOps operator, includes:
    - Tekton pipelines
    - GitHub integration
    - Argo CD
  - Service Mesh
    - istio – open source service mesh
    - Kiali – discovery and integration
    - Prometheus – monitoring
  - Cert Manager
    - Automate certificate management
  - CloudPak For…
    - Data
    - Integration
    - Business Automation

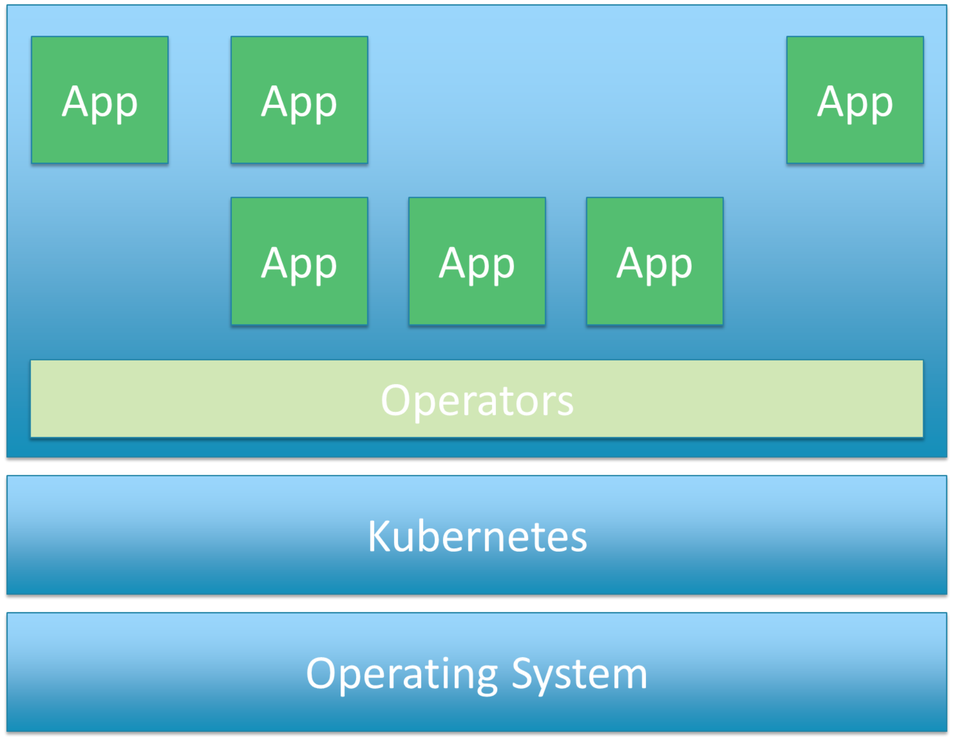


OCP – Cool, but why do I care? (5)

Honest answer?

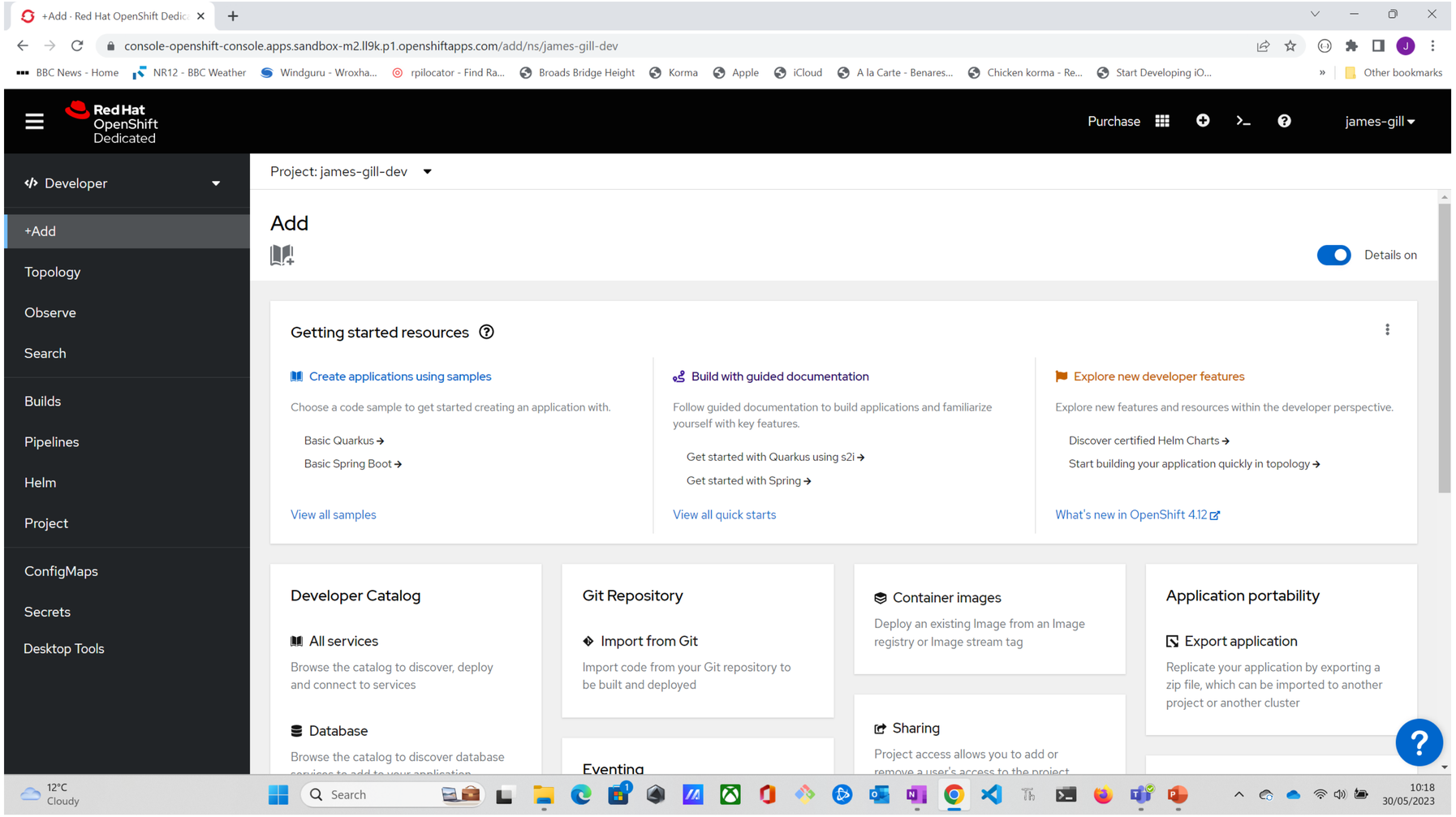
- Your customers will care

- Eh?
- Common developer interface
- Same build and deploy tools regardless of platform
- Same operational monitoring and management tools regardless of platform
  - “ish”
- For Z:
  - Getting younger / integrator supplied developers producing applications on Z
  - Go / VS / GitHub / etc, rather than COBOL, ISPF editor and SCLMs
  - Data gravity
  - Platform quality



OCP – Cool, but why do I care? (6)

Driving it - GUI



Ansible

Why Automate?

Management answers:

- Reliability
- Quality
- The same change methodology through the whole RTL
- Freeing up valuable resources

Technician answers:

- Because we've got too much stuff to do in too little time!
- An automated change is a "standard" change

What is Ansible?

- Ansible is a Python based automation platform
- A control node manages remote systems and their required state
    + Ansible is installed on the control node
    + The target host inventory is on the control node
- Base product is very capable, but can be extended with modules and collections
    + ibm.ibm_zos_core
    + triton.db2_zos_ddf
- Automation is expressed in YAML playbooks with a well defined directory structure.
- Control nodes are (currently) Linux – no fcntl Python package (file locking) on Windows
- Target nodes can be anything with an SSH daemon and Python installed:
    + Linux (x86, s390x, aarch64, etc)
    + z/OS
    + z/VM (via LoZ guest and Feilong SMAPI router)

YAML

```
$ ansible -i hosts all -m ping

host1 | SUCCESS => {
    "ansible_facts":
    { "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong" }

host2 | SUCCESS => {
    "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong" }
```

Assembling an Ansible Framework

On the control node:

- Python
- Ansible
- Any collections you might need – e.g. ibm.ibm_zos_core
- Inventory
- Host variable sets

On the remote systems:

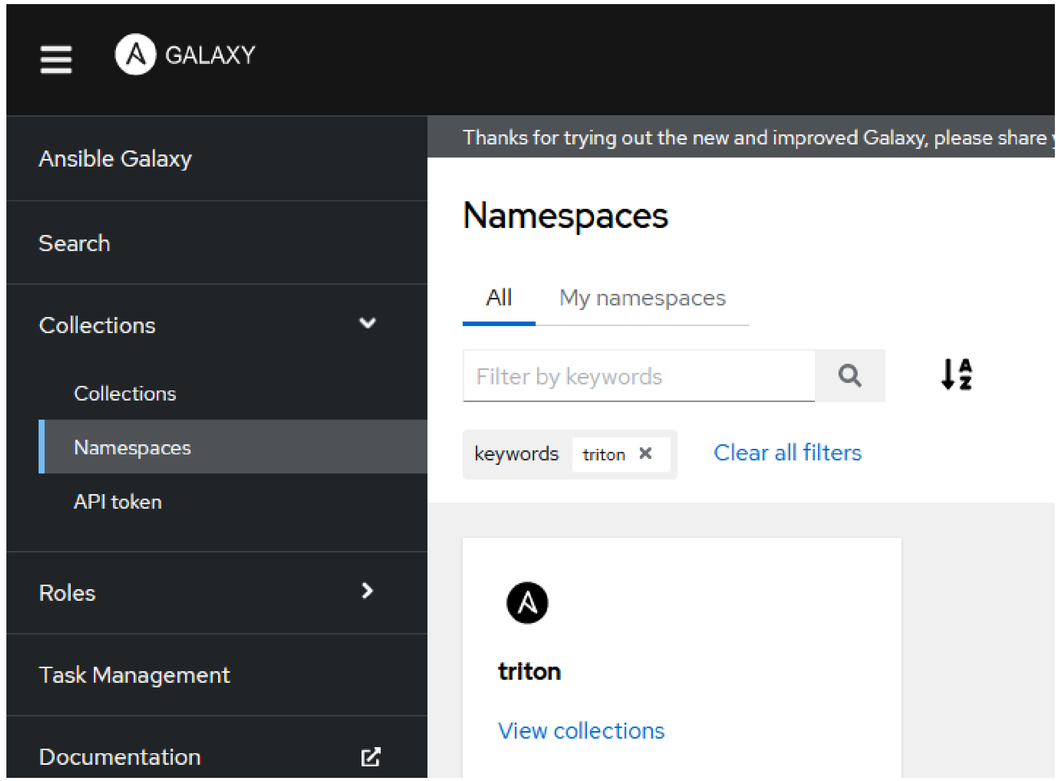
- Python
- Any APIs that you want to support – e.g. ZOA utilities (on z/OS) or OCP CLI (oc), etc.

Extending Ansible to work with Db2

Currently there is no direct support for Db2 for z/OS in Ansible.

Extensions and extending support is largely driven by the community, so:

- modules (in Python) expand capabilities
- collections of modules available on ansible-galaxy
- collections publish into namespaces



GIT Migration?

Why do this?

- Mainframe skills are being lost:
    - staff rotation
    - staff retirement
    - recruitment for “mainframe” resources is difficult – it’s not what the next gen want to do
- This leads to:
    - costly onboarding
    - inadequate skills to support business critical applications hosted on zOS – risk!!
    - the advantages of new tooling can not be realised

Where are we?

What branching strategy do you use?

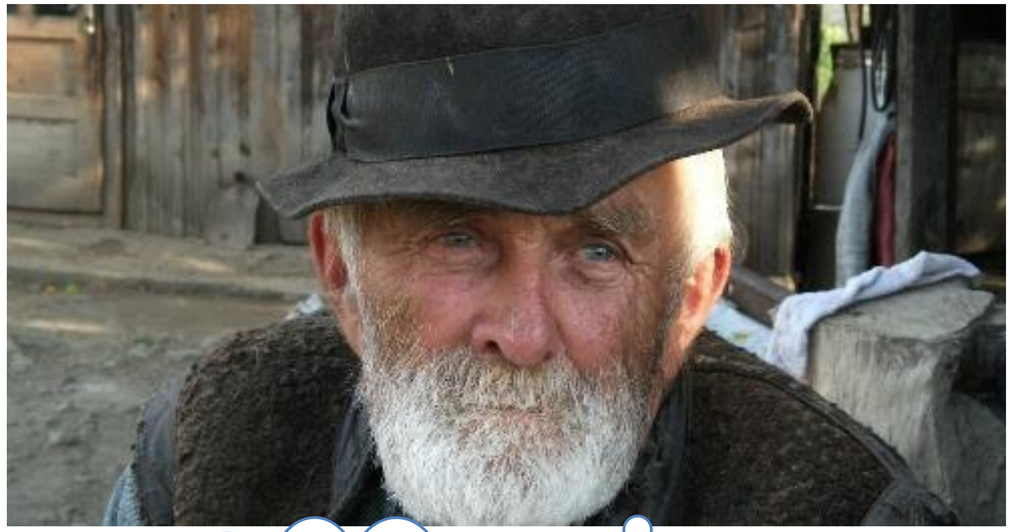
How do you handle merge conflicts without VS Code?

Sometimes I want to fork my repository – How?

What is a delta?

DISCUSSIONS

Where are we?



What is a branching strategy?

You don't get conflicts if you use checkout?

Fork you too!!

What's wrong with a delta? I've been using them since 1982

What branching strategy do you use?

How do you handle merge conflicts without VS Code?

Sometimes I want to fork my repository – How?

What is a delta?

Where are we?

These controls are ridiculous

They just want to stop me being successful

It's all about power

TWO WAY THING

Where are we?

They don't understand the controls we need in a High Street, regulated financial organisation

What RACF groups do you need?

These controls are ridiculous

They just want to stop me being successful

It's all about power

LONG TERM BENEFIT IS TO BRING ZOS TO NEW WORLD

What options are there?

- Your SCLM of choice remains the single source of the truth
- Git is a façade to allow use of VS Code
- The cultural gap is not bridged
- Git is not your really your destination

What options are there?

- Go “Git native”
- Set up a change programme
- The cultural gap can be bridged
- Clear management direction

What is needed?

Processes

- Moving to Git starts with development and ends with migrating code into production:
- Traceability
- Audit

People

- Bring everyone together
- Everyone has something to add
- Teams that bridge the divide

Technology

- Pick tools that really “understand” Git – e.g. DBB
- Consider using zOSMF, Ansible, CMCI etc
- Adopt tools such as Jira

What is needed?

Processes

- Moving to Git starts with development and ends with migrating code into production:
- Traceability
- Audit

People

- Bring everyone together
- Everyone has something to add
- Teams that bridge the divide

Technology

- Pick tools that really “understand” Git – e.g. DBB
- Consider using zOSMF, Ansible, CMCI etc
- Adopt tools such as Jira

Remember, it can be done !!

Start with processes

Environments Team

Change Control Boards

servicenow

Project Managers

Jira

Engineers

Operations

Start with processes

Environments Team

Change Control Boards

servicenow

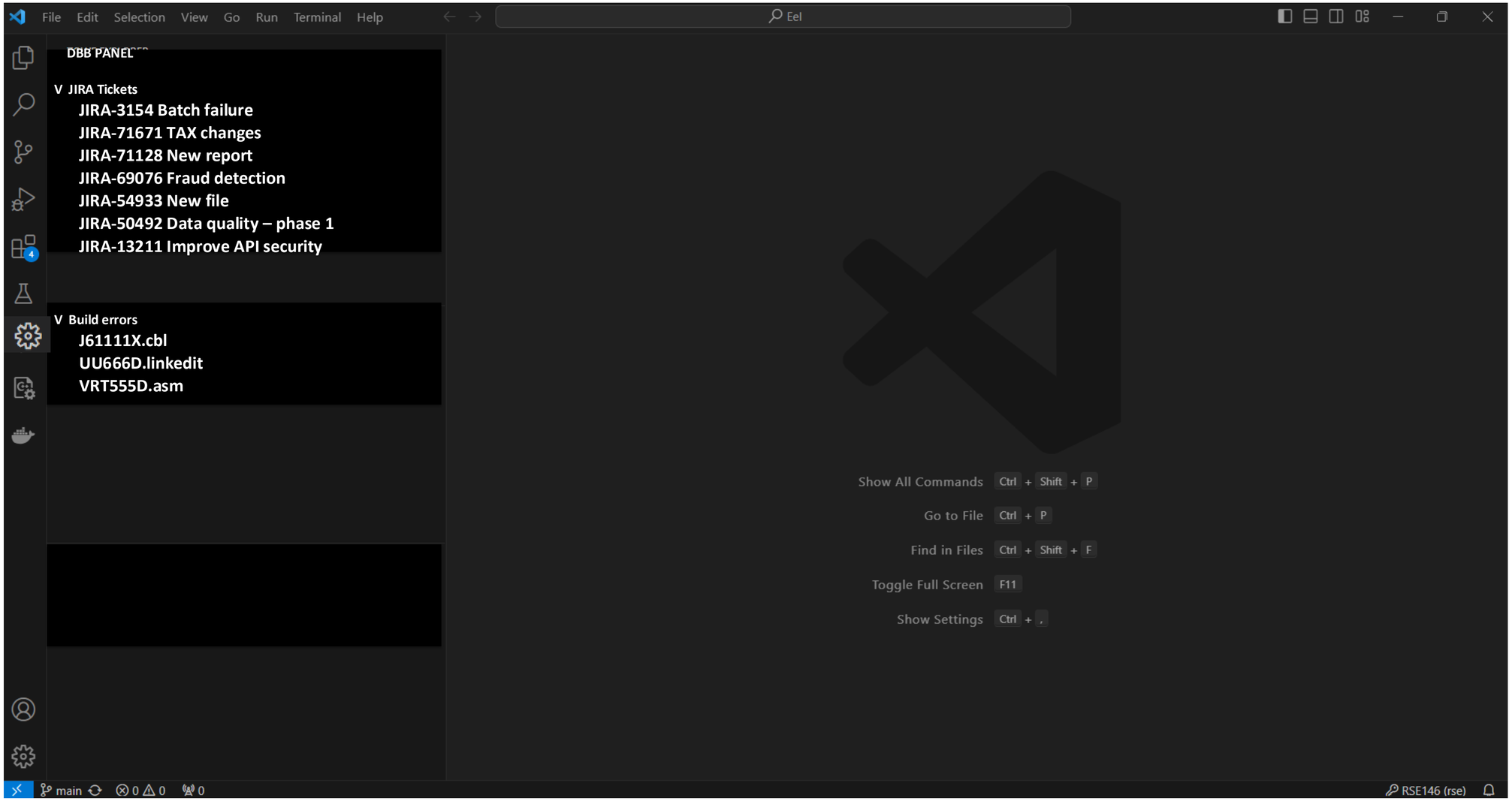
Project Managers

Jira

Engineers

Operations

What's possible?



In summary

- to get onto Git you need to get off what you have today
- it’s not risky
- people will change when they feel comfortable and see the benefit for themselves

- bringing people together needs a structured approach
- start with your processes
- pick tools that understand Git so that they can collaborate effectively

VSAM Migration Project

Situation

- Elderly Transport Department Application in the USA
- New work moved to 3rd party Application on Oracle (boo!!)
- Archive Data resides in VSAM on Mainframe
- Only thing left on the mainframe
- Chosen to move to Db2 on CP4D

High Level plan

- Fixed Price project (ohh no!!)
- 180 VSAM Copy books of Archived Data
- Rexx to read Copybook and strip file accordingly and create CSV file
- Dump CSV files to shared disk
- Python to read CSV file and load into DB2 tables

Actual plan

- Fixed Price - discovery phase only, Fixed price phase 2 once discovery done.
- 180 Copy books – Many had different record types - about 220 record layouts
- Working Assumption was 1 file per Copybook (Customer confirmed this when questioned),,,
  But GDG per day for many years for some Copybooks. 12,000 datasets in all
- Converted REXX to Assembler – Rexx would have run for many years.
- Shared file disks not big enough for datasets ☺ Soooo,
- Restartable Python Load scripts
- Applications still running (even tho turned off ‘many years ago’) moving target ☺

Issues – Mostly Data

- None Valid Dates – 99/99/9999
  12FEB2017
  12-FEB-17, all in same field in same file
  Changes made by different developers to similar code over the years.
- Spaces and letters in number fields
- Random characters – different delimiters required for different GDG Gens of same file.
- Customers Techies getting increasingly grumpy when we were forced to reveal how little they actually knew about their own Apps which they wrote (allegedly).
- Customer had no idea how to query Db2 to check data Migration – so Db2 KT required.

Questions?

22nd February 2024