DevOps and FinOps

A Strategy for Cost-Effective Data Platforms

Morph iT

Hamish Watson

DevOps Consultant, Morph iT Limited

CEO, <FinOps startup>

@theHybridDBA

https://hybriddbablog.com

hamish@morphit.co.nz

- 25 years mucking about with 1s & 0s

- Bringing DevOps to databases (and the masses) is a personal passion

- Understanding data and cloud is a company driver

- Technologist who understands business value...
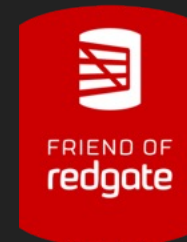
- #makeStuffGo

Microsoft
MVP
Most Valuable
Professional

FRIEND OF
redgate

vmware
vEXPERT
2023

Morph iT

Agenda

- You all know DevOps – quickest recap/intro ever

- Cloud – a platform that enables quick iterative deployment of things

- But that comes at a cost

- So - do you know about FinOps?

Morph IT

Why should we even consider this DevOps thing you talk of?

What DevOps is to me..

https://www.red-gate.com/blog/database-devops/the-four-pillars-of-devops

Tools

Process

People

Culture

The myth of DevOps

+

magic

Tools are cool...

But people are better

OOO Morph iT

Rewards of Digital Transformation using DevOps

- Enhanced Competitiveness and Innovation

- Improved Customer Experience

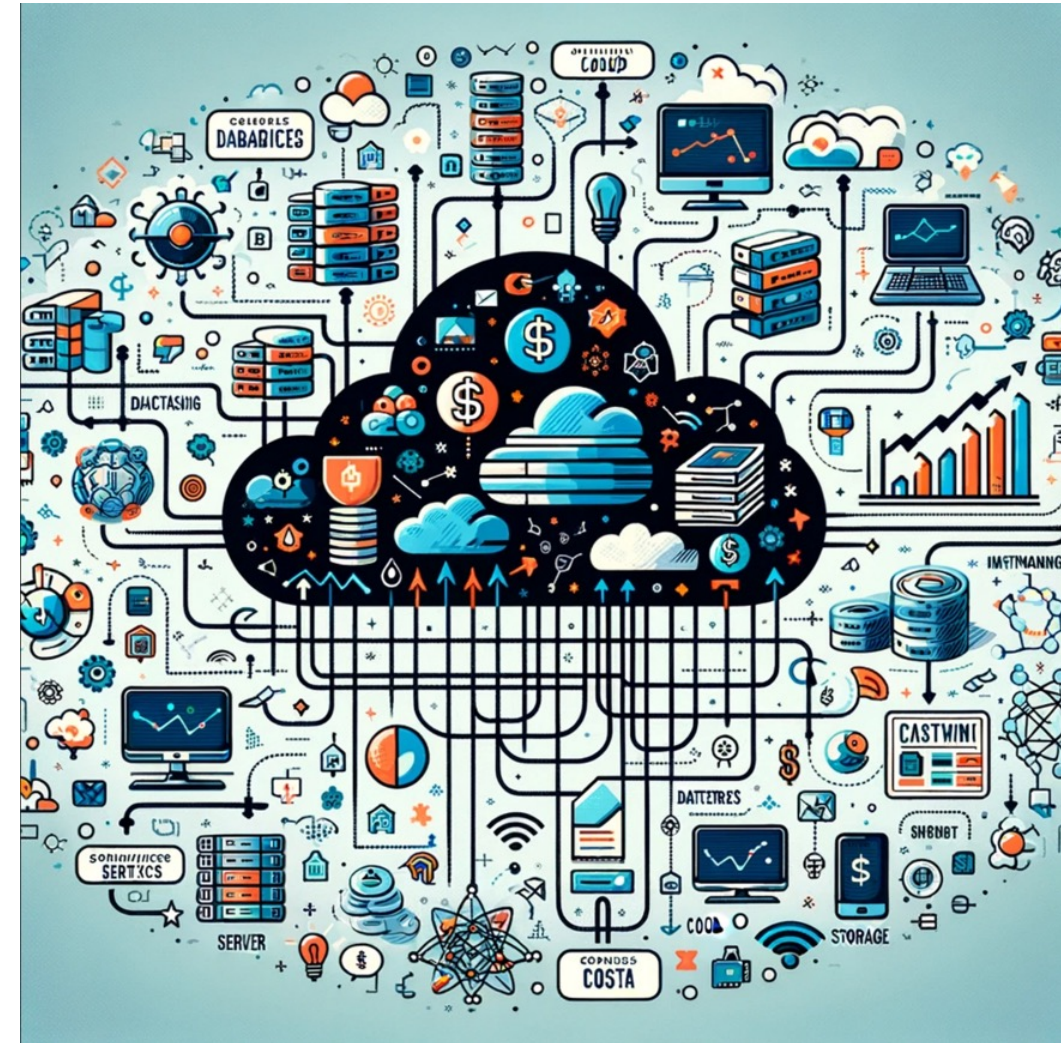- Operational Efficiency and Cost Savings

So what is FinOps??

- Financial Operations

- Cost Optimisation

- Collaboration and Accountability

The role FinOps with cloud-based Data Platforms

- Real-Time Cost Visibility

- Proactive Cost Control

- Cross-Functional Collaboration

Why integrate FinOps with DevOps?

- Scalability and Dynamic Pricing

- Diverse Service Offerings
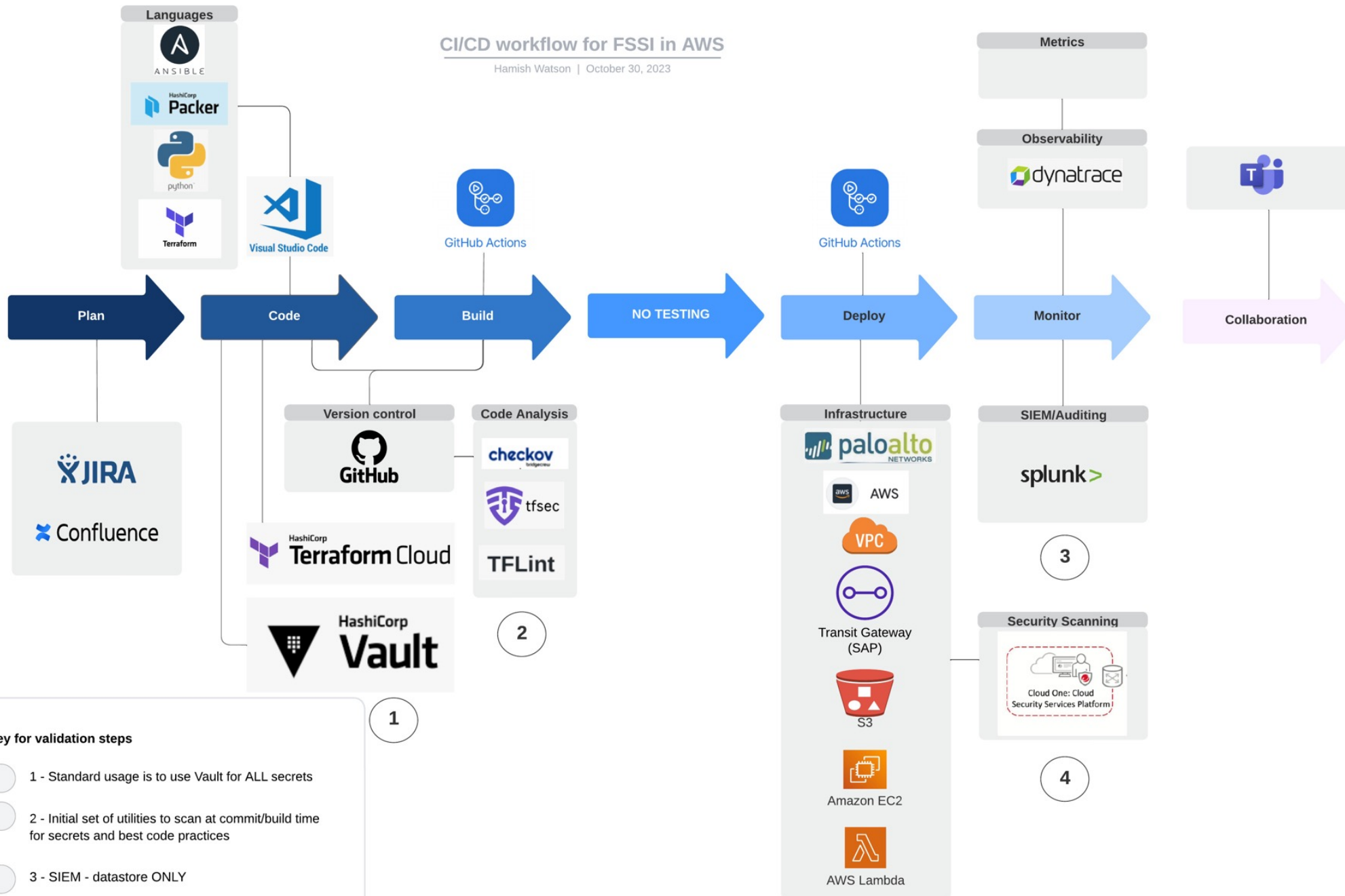
- Complex Architecture and Management

FinOps is the best mate of DevOps

- Enhanced Cost Efficiency

- Improved Financial Visibility

- Faster Innovation with Accountability

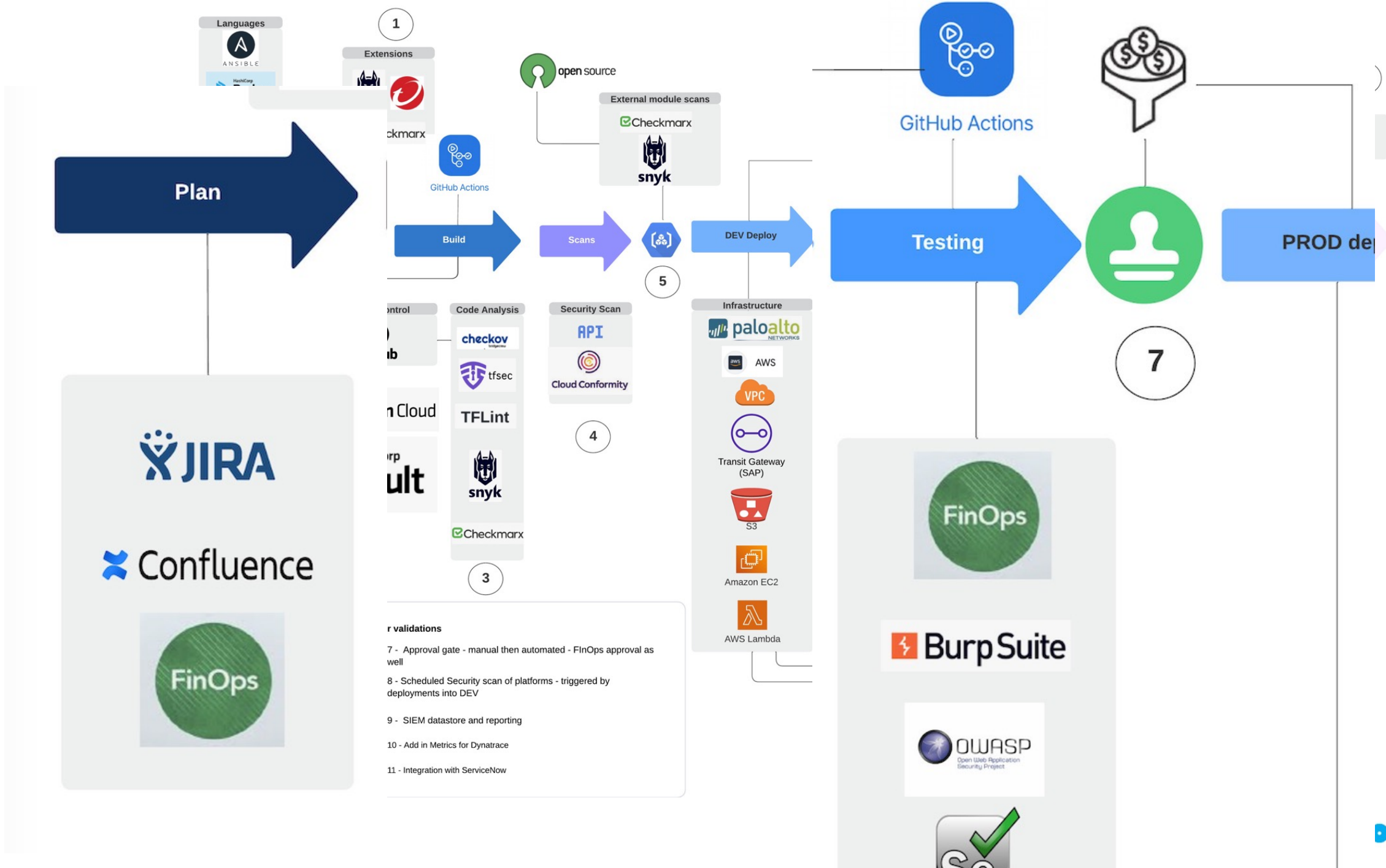# CI/CD workflow for FSSI in AWS

Hamish Watson | October 30, 2023

Languages
- Ansible
- HashiCorp Packer
- Python
- Terraform

Metrics

Observability — dynatrace

Plan → **Code** → **Build** → **NO TESTING** → **Deploy** → **Monitor** → **Collaboration**

Plan
- JIRA
- Confluence

Code
- Visual Studio Code

Build — GitHub Actions

Deploy — GitHub Actions

Version control — GitHub, HashiCorp Terraform Cloud, HashiCorp Vault

Code Analysis — checkov, tfsec, TFLint

Infrastructure — paloalto NETWORKS, AWS, VPC, Transit Gateway (SAP), S3, Amazon EC2, AWS Lambda

SIEM/Auditing — splunk>

Security Scanning — Cloud One: Cloud Security Services Platform

Collaboration — Microsoft Teams

Key for validation steps

1 - Standard usage is to use Vault for ALL secrets

2 - Initial set of utilities to scan at commit/build time for secrets and best code practices

3 - SIEM - datastore ONLY

4 - Security scanning - manual at this stage

MorphIT

The true effect of FinOps and DevOps

- Automated Cost Monitoring

- Proactive Resource Optimisation

- Continuous Deployment with Cost Control

Plan

1
Languages
ANSIBLE
HashiCorp

Extensions
Checkmarx
GitHub Actions

Build

open source

External module scans
Checkmarx
snyk

Scans

DEV Deploy

GitHub Actions

Testing

PROD de...

7

JIRA
Confluence
FinOps

3

Control
ub
Cloud
Vault

Code Analysis
checkov
tfsec
TFLint
snyk
Checkmarx

4

Security Scan
API
Cloud Conformity

Infrastructure
paloalto NETWORKS
AWS
VPC
Transit Gateway (SAP)
S3
Amazon EC2
AWS Lambda

r validations

7 - Approval gate - manual then automated - FInOps approval as well

8 - Scheduled Security scan of platforms - triggered by deployments into DEV

9 - SIEM datastore and reporting

10 - Add in Metrics for Dynatrace

11 - Integration with ServiceNow

FinOps
Burp Suite
OWASP
Open Web Application
Security Project

Morph IT

Tooling to help you with cost management

CASE STUDY

Covid wasn't completely bad….

What this all equated to….

- Client was spending $12M a year in Azure

- I saved them $2M per annum

- That 3-month DevOps project became 3.5 years of work…

- True idempotent environments using terraform

It was so good…..

…. until it wasn't.

- Head of DEV decided to spin up Azure Databricks, used my pipeline template (self service)

- Spun up **ALL THE THINGS**

- $26,000 of storage costs (premium storage for a Proof of Concept)

- Changed pipelines to determine infra cost up front and have a gatecheck

Morph iT

Users > hamish > Dropbox > makeStuffGo > ▼ TF_InfraCosts.tf

```
 1    name: 'Check Terraform Cloud Costs'
 2
 3    on:
 4      push:
 5        branches:
 6          - main
 7      pull_request:
 8        branches:
 9          - main
10    jobs:
11      terraform:
12        runs-on: ubuntu-latest
13
14        - name: Terraform Plan
15          run: terraform plan -out=tfplan
16
17        - name: Terraform Show
18          run: terraform show -json tfplan > plan.json
19
20        - name: Install Infracost
21          run: |
22            curl -sSL https://raw.githubusercontent.com/infracost/infracost/master/scripts/install.sh | sh
23            infracost --version
24
25        - name: Infracost breakdown
26          run: infracost breakdown --path=plan.json --format=json --out-file=cost.json
27
28        - name: Post Infracost comment
29          uses: infracost/actions/comment@v1
30          with:
31            path: cost.json
32            behavior: update
33
```

FOR SOME AMAZING KIWI CHOCOLATES:

What is missing from this code?

○○●Morph iT

APPROVAL STEP

```yaml
approval:
    needs: terraform
    runs-on: ubuntu-latest

    steps:
    - name: Request approval
      id: approval
      uses: chetan/inaction@v1.0
      with:
        GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        APPROVE_MESSAGE: "Please review and approve the cloud cost estimate."
        DENY_MESSAGE: "Cloud cost estimate not approved. Please check the costs."
        APPROVAL_LABELS: "awaiting-approval,approved,denied"
        REVIEWERS: "heymo0sh69" # Actual GitHub username of the reviewer
        CUSTOM_COMMENT: "Cloud cost estimation for the changes."

    - name: Check approval
      run: |
        if [ "${{ steps.approval.outputs.action }}" != "approved" ]; then
          echo "Action not approved."
          exit 1
        fi
      id: check_approval

deploy:
  needs: approval
  runs-on: ubuntu-latest

  steps:
  - name: Checkout code
    uses: actions/checkout@v2

  - name: Set up Terraform
    uses: hashicorp/setup-terraform@v1
    with:
      terraform_version: 1.0.11 # specify your desired Terraform version

  - name: Terraform Apply
    run: terraform apply --auto-approve tfplan
```
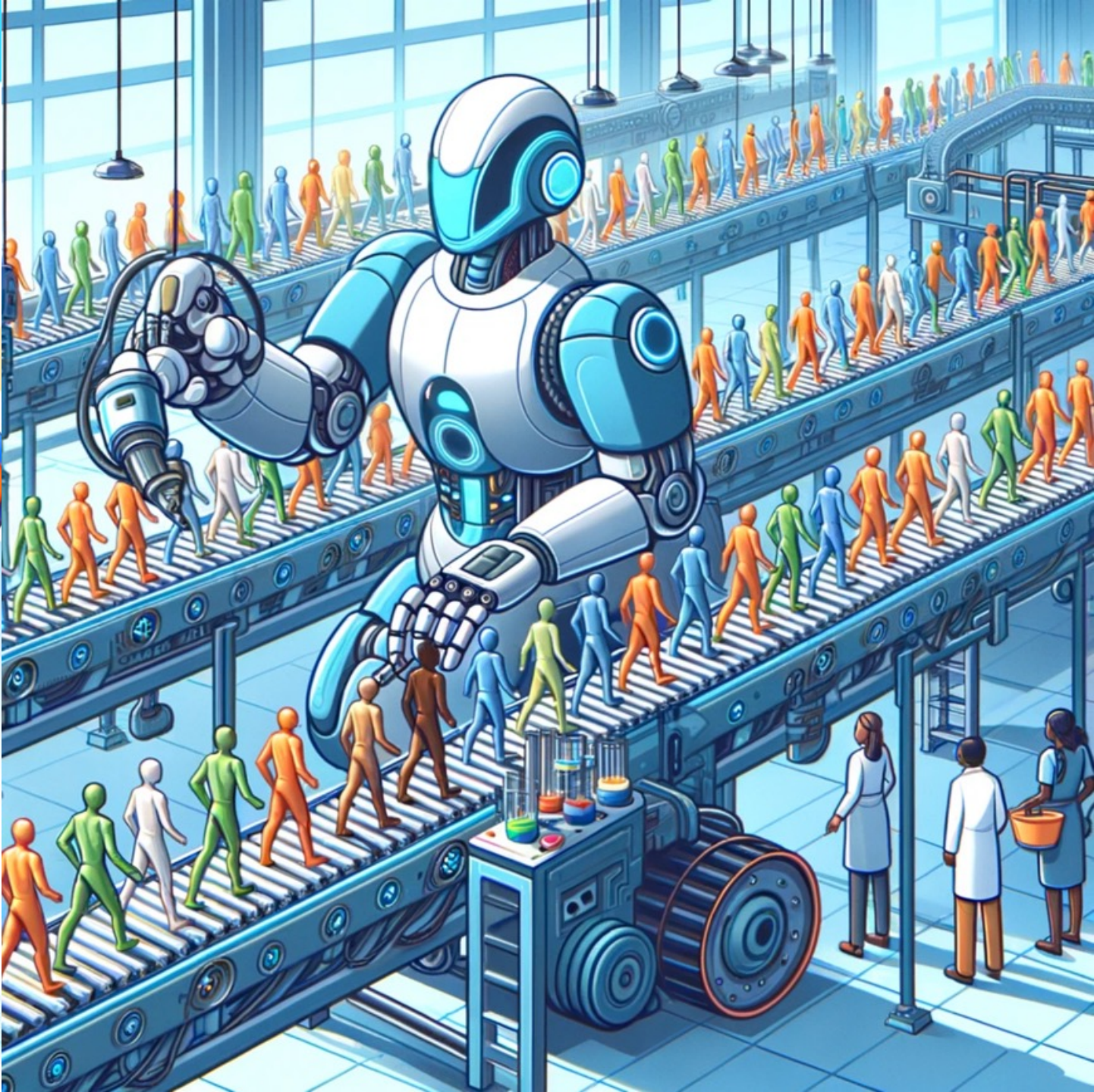
Morph iT

Summary of all this

- Enhanced Cost Management

- Improved Operational Efficiency

- Collaborative Culture

- Continuous Innovation with Cost Control

OK… one more thing

- 2024 is the year that we should all be doing **Test Data Management**
- FinOps can help with this..
- **Cost Visibility and Tracking**
- **Cost Optimization**
- **Governance and Compliance**
- **Dynamic Scaling and Flexibility**
- **Collaboration and Accountability**
- Test Use Case (yes this is a pun)
- Check out the **Redgate** booth – they have free stuff

Morph IT

Questions?

THANK YOU
Have an awesome DataGrillen!!

Hamish Watson

@theHybridDBA

https://hybriddbablog.com

hamish@morphit.co.nz

Morph IT