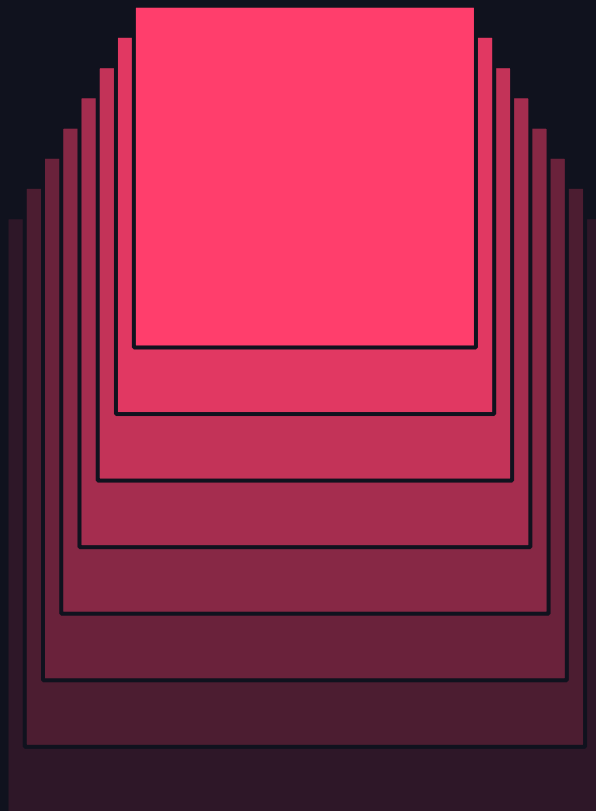


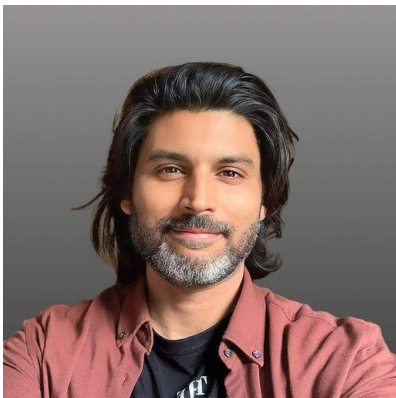
Path to Production:

CICD for Seamless Inner to
Outer Dev Loops



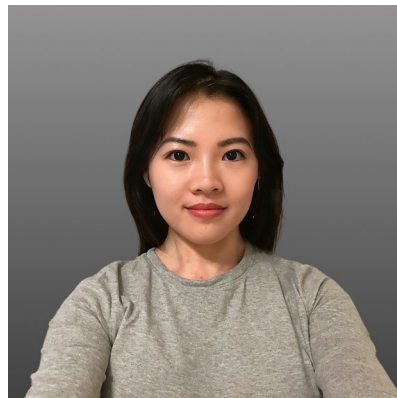
Nicole Lu and Saad Ansari (Product Managers)
Thursday, June 13 2024

Intros



Saad Ansari

- Product Manager for Developer Ecosystem and Databricks Workflows



Nicole Jingting Lu

- Product Manager for Git Connectivity and Productionization at Databricks
- Managed Industry Solution Accelerators at Databricks

From POC to Production

Common challenges faced by data teams

Data Quality and consistency variation across dev and prod - will your code run successfully?

Monitoring and maintenance is challenging if you cannot debug your code to root cause prod issues

Collaboration and sharing can create code conflicts - you need tools and process to review and collaborate

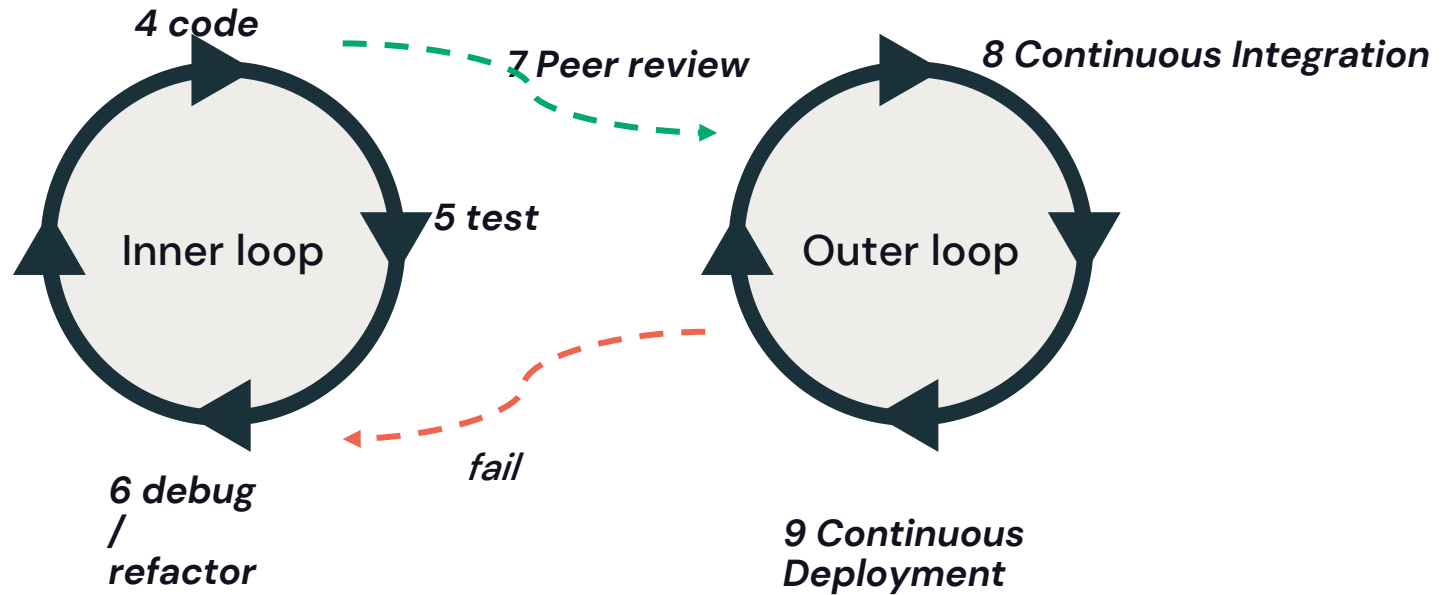


Software best practices ... what are they?

1 setup Dev environment

2 source control

3 setup project

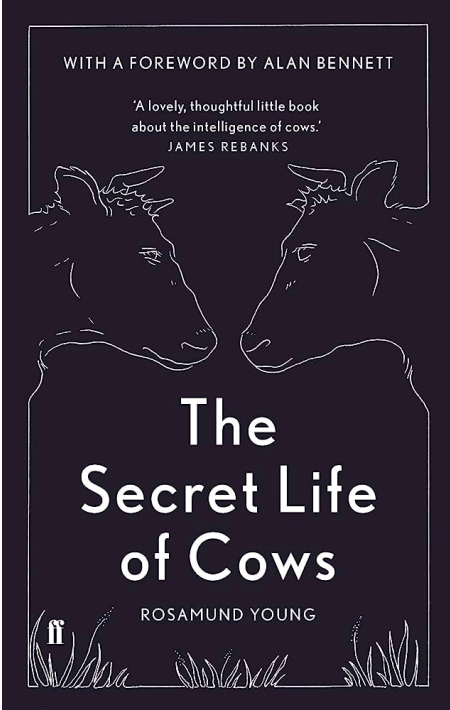
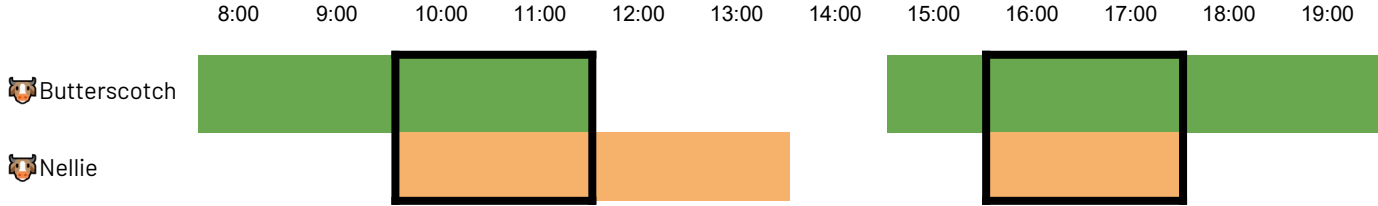


Cows making friends

Reference: *The Secret Life of Cows* by Rosamund Young

Cows have best friends and they get stressed when they are separated

Hypothesis: Given that cows are social and form friendships, do cow BFFs take their meals together



Demo

Let's look at the data ...



Demo

Git

Refactor

Debug

Test

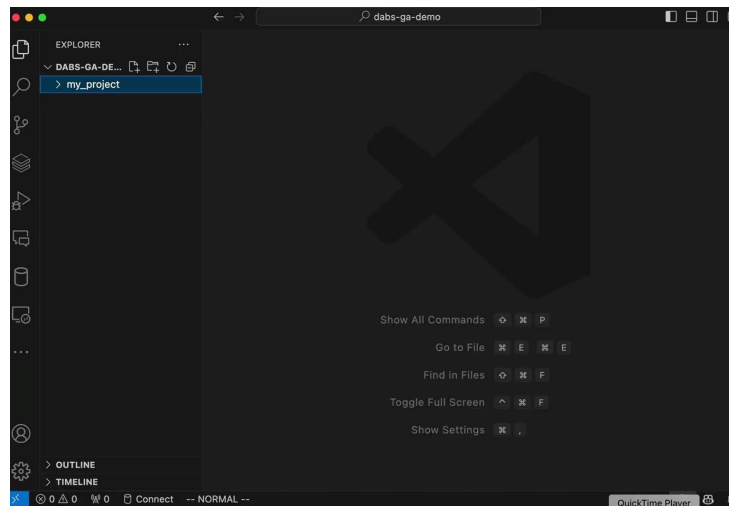


Why use Databricks Asset Bundles (DABs)?

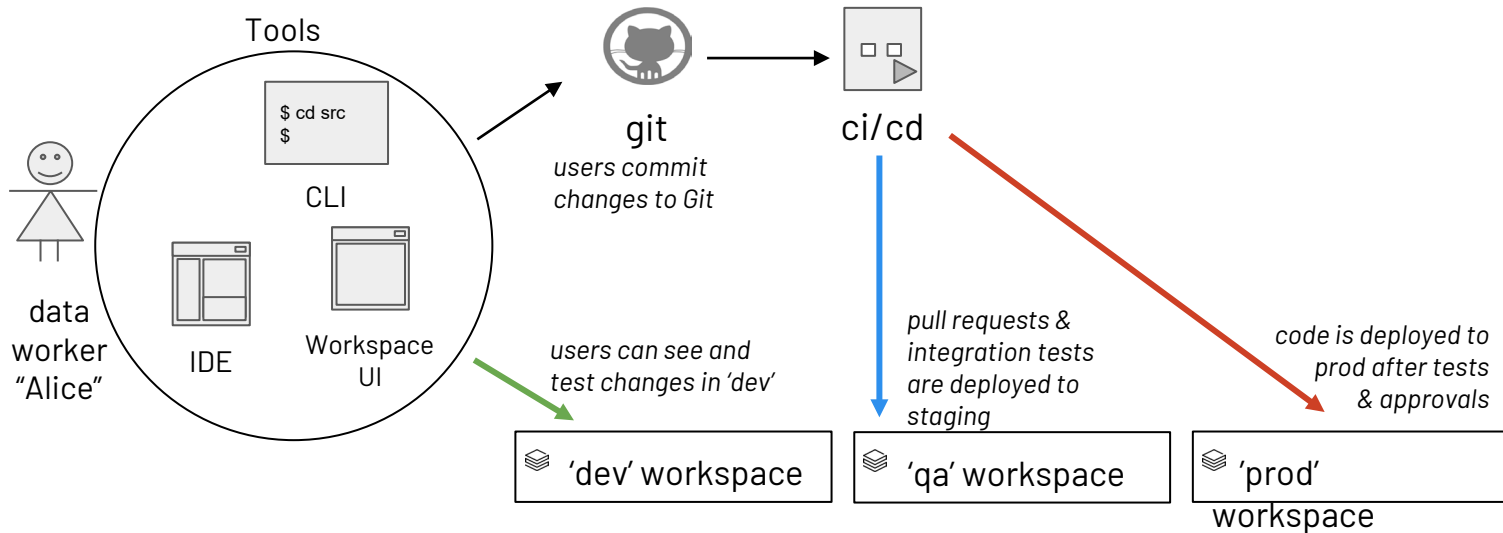
- Bundle resources like jobs, pipelines, notebooks so you can **version, test, and deploy your project as a unit**
- Adopt **software engineering best practices**: facilitate source control, code review, testing, and continuous integration and delivery (CI/CD)
- **Isolate development copies of the project** so code and configuration changes can be tested without impacting production
- **Eliminate manual deployment, intervention and validation**
- **Improve developer productivity** by avoiding fire drills in production, streamlining workflow, fostering collaboration
- **Define consistent configuration** across development, staging, and production environments
- **Make deployments repeatable and changes auditable**

What are DABs?

- Declarative format for describing resources and code (yaml)
- Override settings by environment (i.e. pause jobs in dev by default)
- Set granular permissions on your assets across your project
- Use variables and lookups for modular configuration
- Build and deploy shared code and libraries (i.e. Python wheels)
- Use the Databricks CLI to deploy across environments
- Automate your deployments using Github Actions, Azure DevOps, Jenkins or the CI/CD tool of your choice



"Write once, deploy everywhere"



```
→ databricks bundle deploy -t "dev"
→ databricks bundle run pipeline -refresh-all -t "dev"
→ databricks bundle deploy -t "qa"
→ databricks bundle run pipeline -refresh-all -t "qa"
→ databricks bundle deploy -t "production"
→ databricks bundle run pipeline -refresh-all -t "production"
```

Demo

Databricks as Code

CICD