

Databricks: DataOps – Bundles, Terraform and CI/CD

Databricks: DataOps – Bundles, Terraform and CI/CD

is a hands-on workshop focused on deployment automation in Databricks, delivered **exclusively as a dedicated training for organizations (private groups only)**.

The program focuses on how to promote jobs and data pipelines across environments in a repeatable, controlled, and reliable manner using **Declarative Automation Bundles** (formerly **Databricks Asset Bundles**), **Git**, and **CI/CD pipelines**.

Participants go through the entire deployment lifecycle: from defining the division of responsibilities between **Terraform** (platform resources) and **Bundles** (data artifacts), to designing the bundle structure, implementing tests and parameters, deploying to an isolated development environment, and designing a promotion process to testing and production environments.

The workshop is delivered using a repository and environment agreed upon with the customer prior to the training. As a result, participants work with a setup that closely resembles their own organizational environment and deployment processes.



Training recipients

The training is intended for teams that deploy data solutions in Databricks and want to automate this process.

The course is particularly suitable for:

- data engineers preparing jobs and pipelines for deployment,

- DevOps and platform engineers responsible for Databricks environments,
- technical leads setting the rules for promoting changes between environments,
- teams that currently deploy changes manually and want to move to CI/CD.

The training will be especially valuable for organizations that already have working processes in Databricks and need a safe, repeatable way to deploy them.



Benefits

- Division of responsibility – you will learn what belongs to Terraform and what to Declarative Automation Bundles, and how platform and data teams share the work.
- Declarative Automation Bundles – you will learn to build a bundle structure with targets, variables and per-environment overrides.
- Deploying to dev – you will run bundle validate, bundle plan and bundle deploy on an isolated dev environment.
- A CI/CD pipeline – you will design a pipeline triggered by a pull request, with validation, tests and deployment.
- Promotion between environments – you will learn the criteria and approvals for moving from dev to test and prod.
- Deployment security – you will learn to use a service principal, run_as and secrets, and to review changes in audit logs.



Training program

. Module – Terraform and Bundles: who is responsible for what

Module goal: the participant knows which parts of the platform the platform team deploys, which the data team deploys, and with which tools.

- Terraform – setting up and configuring the platform: workspaces, catalogs, compute policies, permissions
- Declarative Automation Bundles (formerly Databricks Asset Bundles) – deploying what the data team builds: jobs, pipelines, dashboards, Genie Agents, models
- Identities – user accounts, technical accounts (service principals) and groups; who can deploy to which environment
- Signing in deployment tools without passwords in code (OAuth, identity federation for CI)
- Separating dev / test / prod environments – separate workspaces or separate catalogs
- Lab: participants map out the division of responsibility for their organization (discussion, approx. 10 min)

2. Module – Declarative Automation Bundles

Module goal: the participant describes a job or pipeline as code and deploys it with a command.

- What a bundle is – the databricks.yml configuration file and the code in the repository, which together describe the solution
- Project starter templates (databricks bundle init)
- Target environments (targets) – the same resources deployed to dev and prod with different settings
- Variables and overrides – for example, a different catalog, compute size or schedule in each environment
- Workflow: checking the configuration (validate), previewing changes (plan), deploying (deploy), running (run)
- Deploying a bundle directly through the API (the direct deployment engine) – the only engine available since CLI 1.20.0; older Terraform-based deployments are migrated automatically on deploy
- Lab: the participant describes a job and pipeline from the repository as a bundle and checks the deployment plan on dev

3. Module – Tests, parameters and promotion rules

Module goal: the participant knows how to check a change before it moves to the next environment.

- Unit tests for transformation code – run locally or in CI (pytest; Databricks Connect installed before the training)
- Data tests after deployment – quality rules and control queries
- Environment-specific parameters – catalog, schema, compute, schedule
- Criteria for moving dev → test → prod and who approves the change
- Lab: the participant writes a test for a transformation function and parameterizes the catalog per environment

4. Module – CI/CD for Databricks

Module goal: the participant designs an automated process for checking and deploying changes.

- Working with Git – branches, pull requests and code review; Git folders in the workspace
- CI pipeline – on every pull request: tests, bundle validation and a preview of changes
- CD pipeline – after approval: deployment to test, and after sign-off to prod
- GitHub Actions or Azure DevOps – configuration, secrets and signing in to Databricks
- Lab: the participant designs a CI/CD pipeline for the workshop repository

5. Module – Workshop: repository, dev environment and deployment

Module goal: the participant carries out a full deployment to the dev environment on their own.

- Workshop: using an agreed repository, participants set up the dev environment, check the configuration and plan, deploy a job or pipeline, run it and verify the result. The outcome is a working, repeatable deployment to the dev environment.

6. Module – Terraform for platform resources

Module goal: the participant understands how the platform team manages Databricks configuration as code.

- The Databricks provider for Terraform – configuration at the account and workspace level
- Sample resources – catalogs, schemas, permissions, compute policies
- Configuration state, previewing changes and detecting manual changes made outside Terraform
- When a resource belongs in Terraform and when in a bundle (demo)

7. Module - Deployment security and the further promotion plan

Module goal: the participant secures the deployment process and plans the next steps.

- Secrets – storing passwords and keys outside the code (secrets in Unity Catalog with permissions managed like those for tables, Databricks secret scopes, in Azure also backed by Key Vault; secrets in GitHub or Azure DevOps for CI pipelines)
- Running jobs under a technical account (run_as) and permissions on deployed resources
- Audit logs – who deployed what and when
- Security checklist and the plan for promotion to test and prod

Out of scope: network provisioning, cloud identities (IAM, Entra ID), private endpoints and landing zones, and deployments to test and prod environments during the course. These topics need a separate architecture workshop.



Expected preparation of the participant

- Practical knowledge of Git and working in a terminal
- Completion of DBX-LKF (Lakeflow) or a working job or pipeline in Databricks that is to be deployed
- Before the training: an agreed repository, a service principal with minimal permissions and an isolated dev environment
- Databricks CLI and Databricks Connect installed (we send instructions before the training)

To increase the comfort of work and training's effectiveness we suggest using an additional monitor. The lack of additional monitor does not exclude participation in the training, however, it significantly influences the comfort of work during classes.

You can find information and requirements of participation in Distance Learning trainings at:

<https://www.altkomakademia.pl/en/how-it-works/about-courses/>



Training Includes

- access to Altkom Akademia student

Training method:

- agreement on the scope, repository and environment before the training
- a workshop on a repository and dev environment agreed with the client
- exercise materials in electronic form

The environment and cloud resources are provided by the Client.



Duration

1 days / 7 hours

Language

- Training: English
- Materials: English