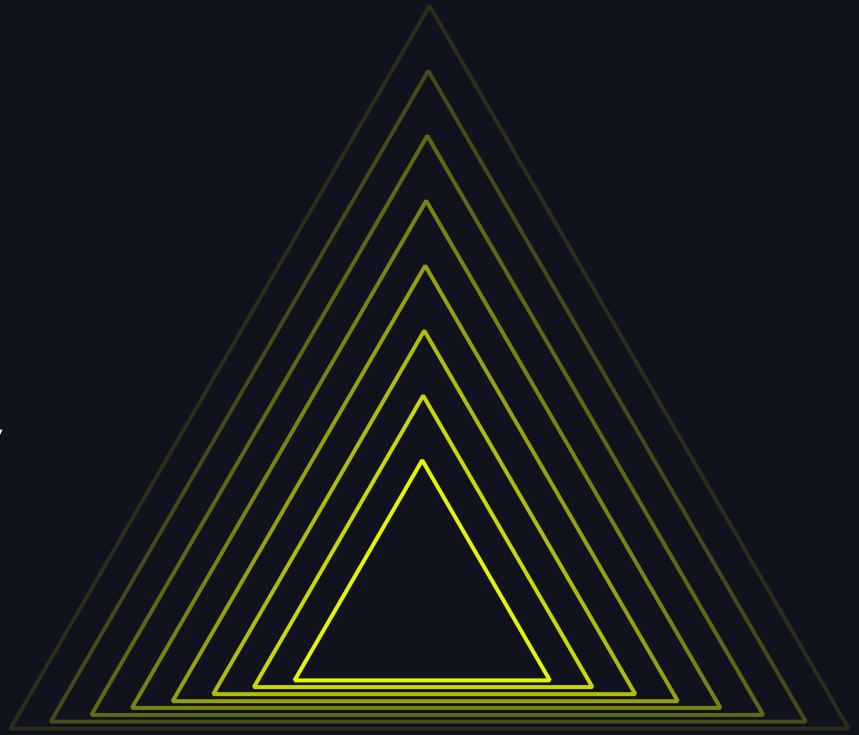
DATA+AI SUMMIT BY databricks

Robust Python apps on Databricks: UCX case study



Serge Smertin

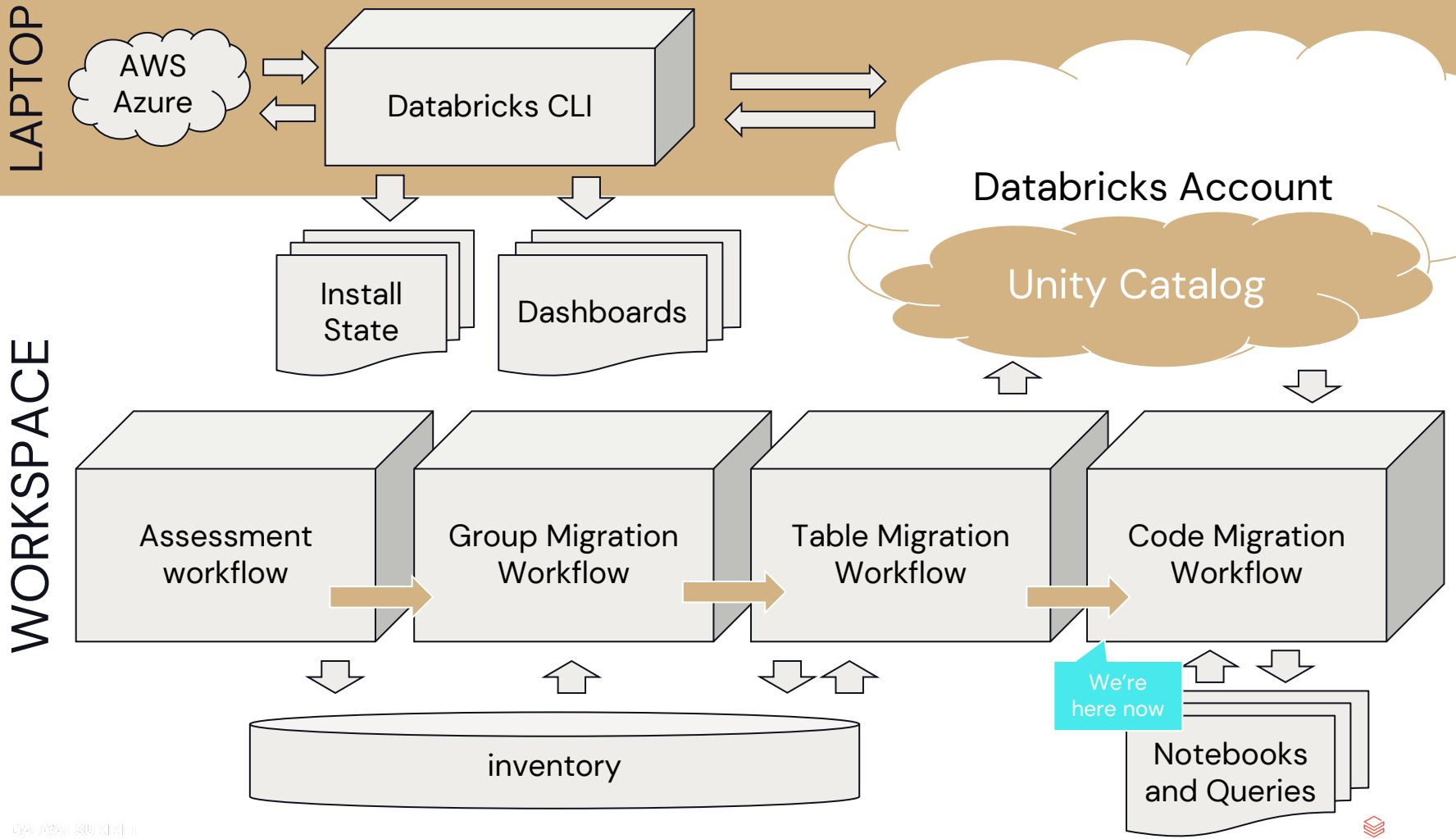
DATA+AI SUMMIT BY databricks

WHAT PROBLEMS DID WE HIT, SO YOU DON'T HAVE TO.

About Serge

- Using Apache Spark since ~2015
- At Databricks since 2019
- Created Databricks Terraform Provider
- Author of Databricks SDKs
- Driving Databricks Labs
- Years in cybersecurity and payments before that

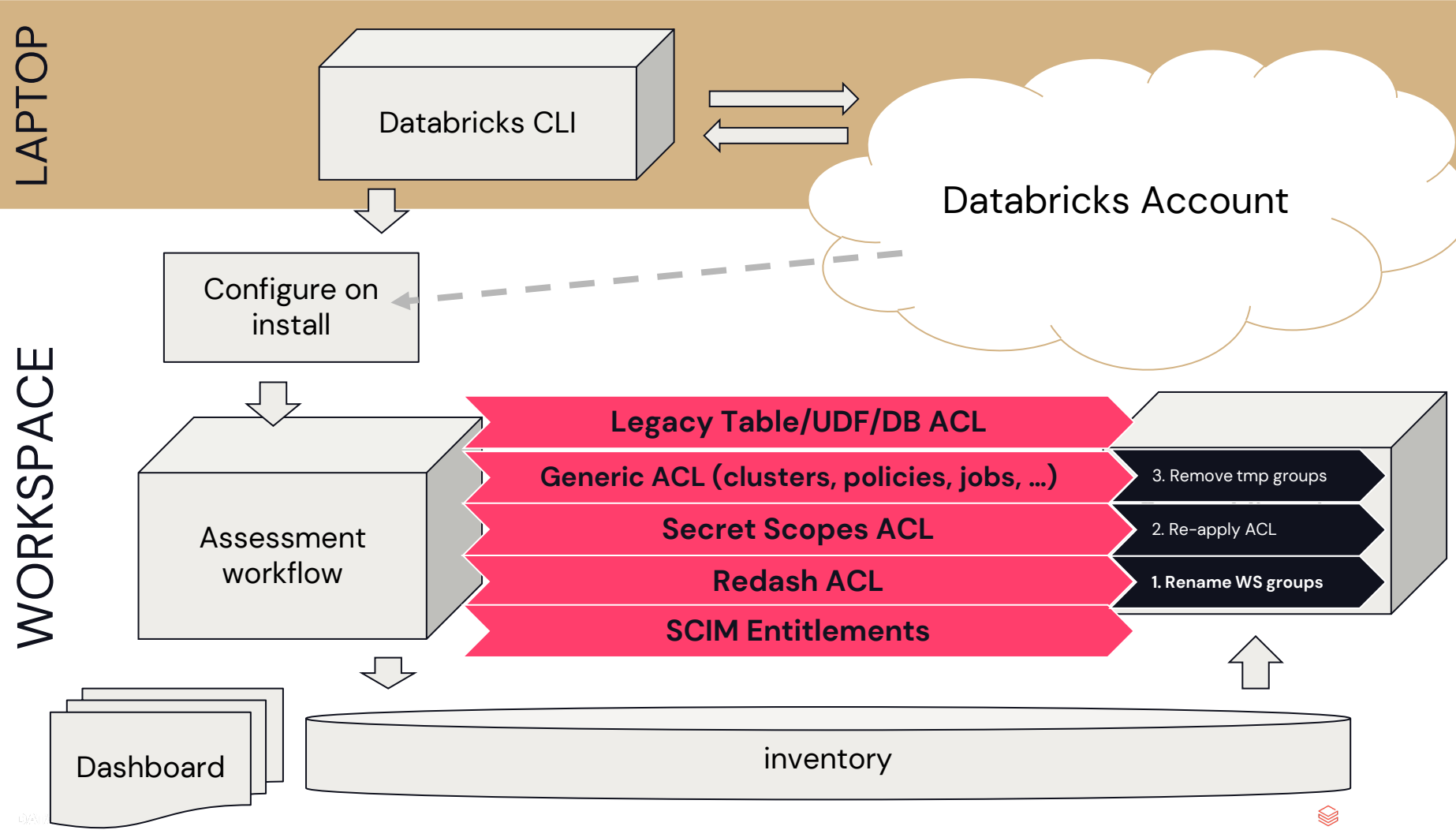
- OVERVIEW
- GROUP MIGRATION
- TABLE MIGRATION
- CODE MIGRATION


LAPTOP
AWS
Azure
Databricks CLI
Databricks Account
Unity Catalog
Install
State
Dashboards
WORKSPACE
Assessment
workflow
Group Migration
Workflow
Table Migration
Workflow
Code Migration
Workflow
We're
here now
inventory
Notebooks
and Queries

```
→  ~ databri
```


ucx
Automated upgrade to Unity Catalog

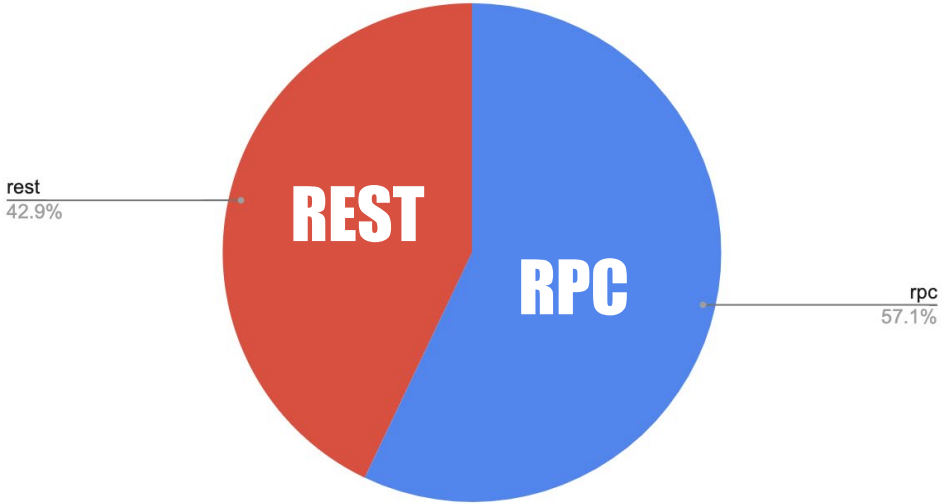
GROUP MIGRATION


LAPTOP
Databricks CLI
Databricks Account
Configure on install
WORKSPACE
Assessment workflow
Legacy Table/UDF/DB ACL
Generic ACL (clusters, policies, jobs, ...)
Secret Scopes ACL
Redash ACL
SCIM Entitlements
3. Remove tmp groups
2. Re-apply ACL
1. Rename WS groups
Dashboard
inventory

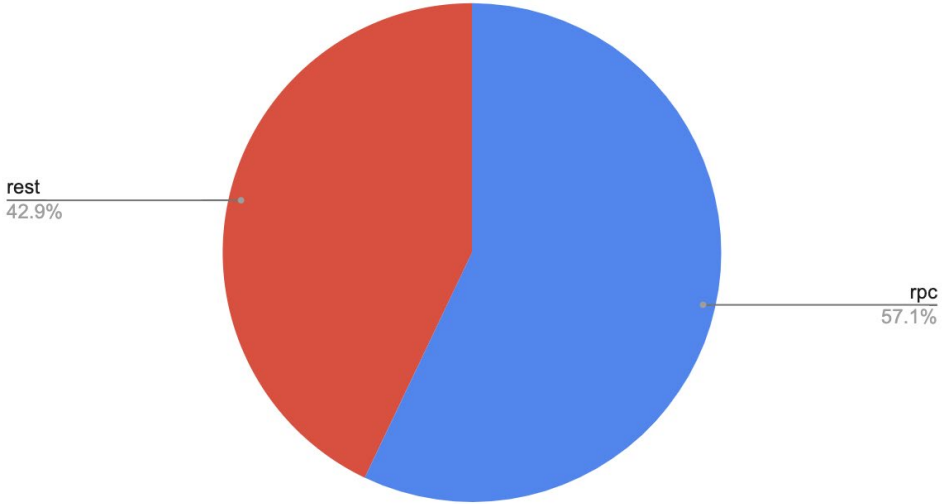
CHALLENGES:
-CALLING API
-LOGGING
-ERROR RECOVERY

DIRECT API INTEGRATION IS HARD

API Path style

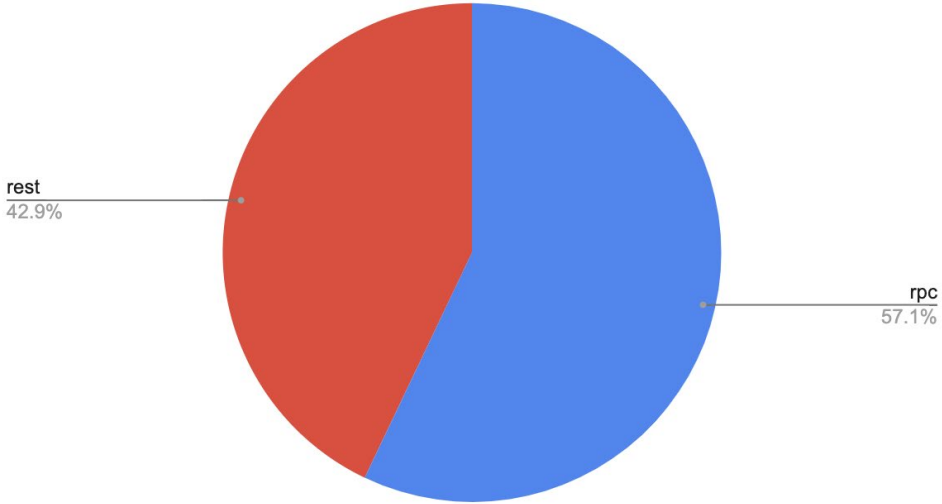


API Path style



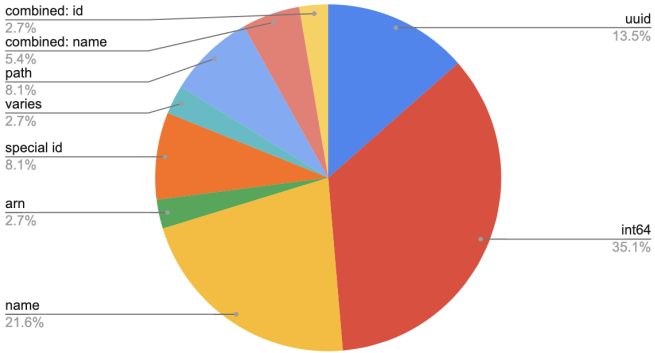
/api/2.0/git-credentials	rest
/api/2.0/git-credentials/{credential_id}	rest
/api/2.0/global-init-scripts	rest
/api/2.0/global-init-scripts/{script_id}	rest
/api/2.0/instance-pools/create	rpc
/api/2.0/instance-pools/delete	rpc
/api/2.0/instance-pools/edit	rpc
/api/2.0/instance-pools/get	rpc
/api/2.0/instance-pools/list	rpc
/api/2.0/ip-access-lists	rest
/api/2.0/ip-access-lists/{ip_access_list_id}	rest

API Path style

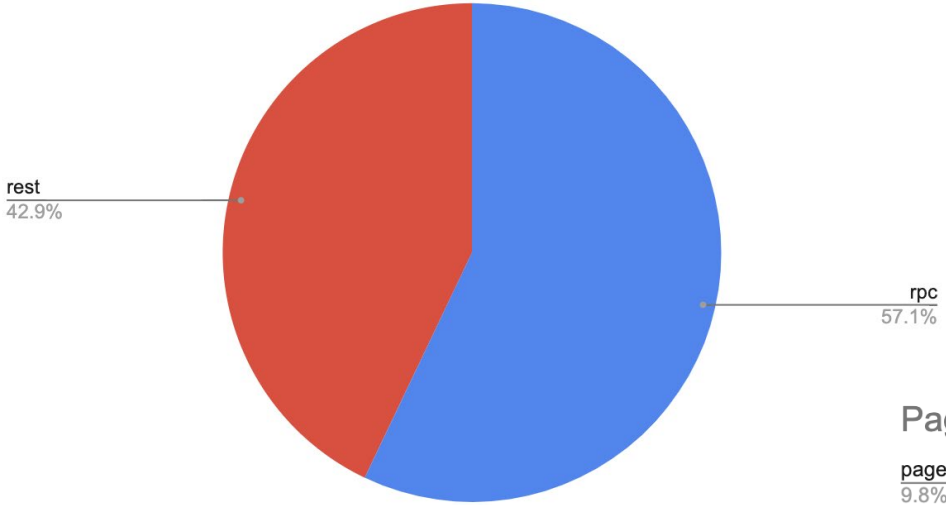


/api/2.0/git-credentials	rest
/api/2.0/git-credentials/{credential_id}	rest
/api/2.0/global-init-scripts	rest
/api/2.0/global-init-scripts/{script_id}	rest
/api/2.0/instance-pools/create	rpc
/api/2.0/instance-pools/delete	rpc
/api/2.0/instance-pools/edit	rpc
/api/2.0/instance-pools/get	rpc
/api/2.0/instance-pools/list	rpc
/api/2.0/ip-access-lists	rest
/api/2.0/ip-access-lists/{ip_access_list_id}	rest

Identifier formats across Terraform resources

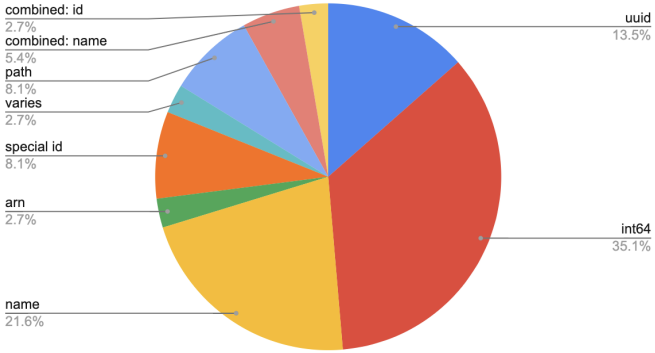


API Path style

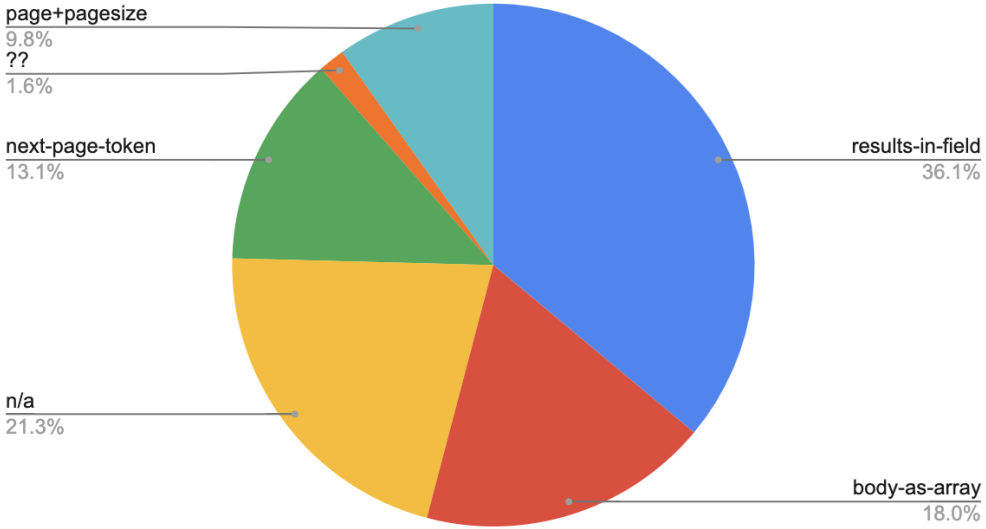


/api/2.0/git-credentials	rest
/api/2.0/git-credentials/{credential_id}	rest
/api/2.0/global-init-scripts	rest
/api/2.0/global-init-scripts/{script_id}	rest
/api/2.0/instance-pools/create	rpc
/api/2.0/instance-pools/delete	rpc
/api/2.0/instance-pools/edit	rpc
/api/2.0/instance-pools/get	rpc
/api/2.0/instance-pools/list	rpc
/api/2.0/ip-access-lists	rest
/api/2.0/ip-access-lists/{ip_access_list_id}	rest

Identifier formats across Terraform resources



Pagination styles



USE DATABRICKS SDK (FOR PYTHON) TO SAVE MONTHS OF EFFORT.

Development

Authenticate through Databricks CLI, Azure CLI, Visual Studio Code, or in Databricks Notebooks.

```
$ databricks auth login
```

```
$ az login
```

Production or CI

Authenticate through environment variables. Leverage Kubernetes secrets and/or CI runner secret redaction.

```
$ export DATABRICKS_HOST=...
$ export ARM_CLIENT_ID=...
$ export ARM_TENANT_ID=...
$ export
ARM_CLIENT_SECRET=...
$ python3 run-app.py
```

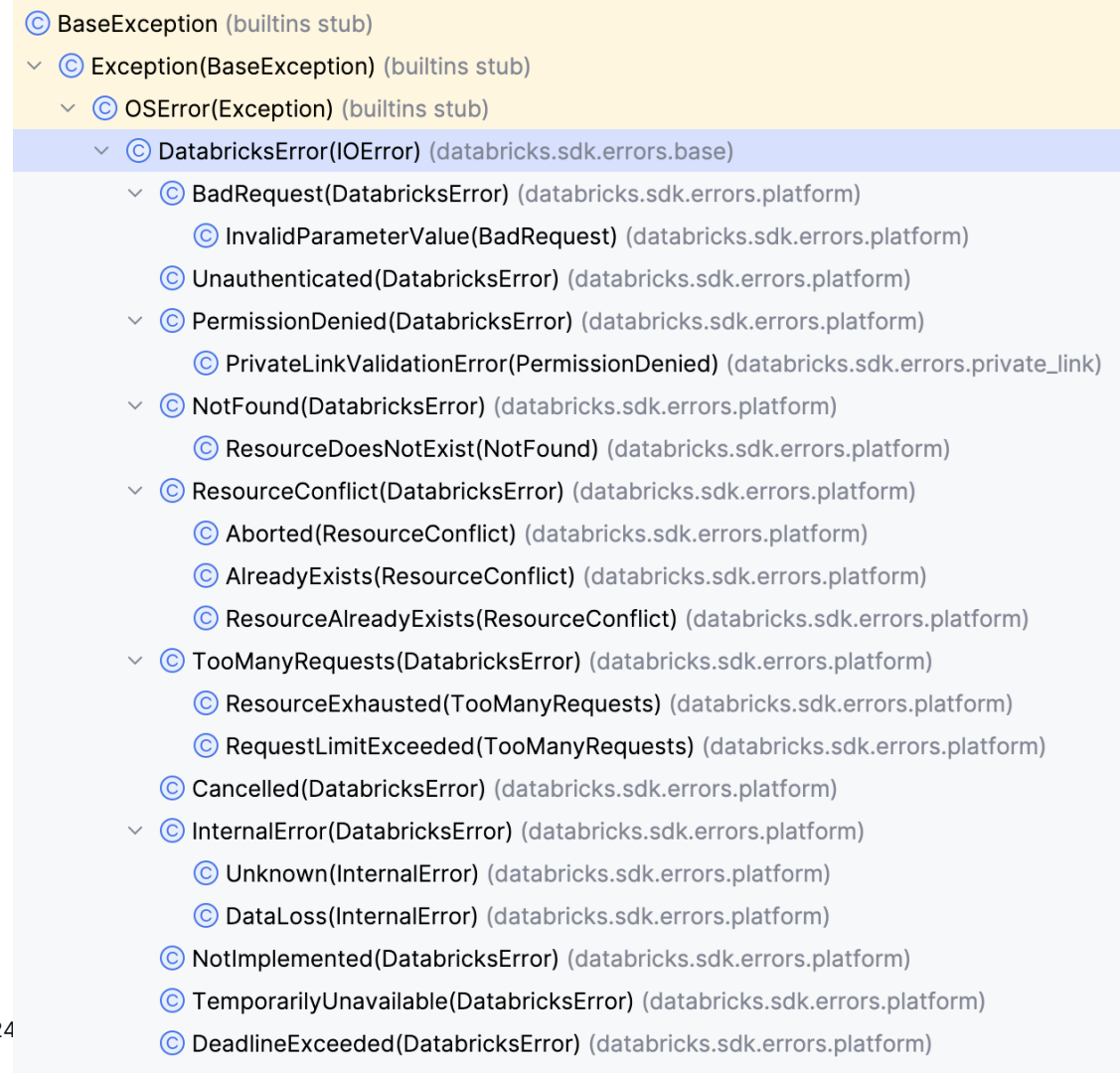
```
from databricks.sdk import WorkspaceClient
w = WorkspaceClient()
```

DATABRICKS SDK FOR PYTHON

ERROR HIERARCHY

Catch the right thing in the right place to recover properly

- Consistent across all SDK
- Exceptions are named after Databricks error codes
- Inheritance is modelled after HTTP status codes



EVENTUALLY CONSISTENT API

First attempts are not always successful

PYTHON

```python
@retried(on=[InternalError, ResourceConflict, DeadlineExceeded])
@rate_limited(max_requests=35, burst_period_seconds=60)
def _delete_workspace_group(self, group_id: str, display_name: str) -> None:
    try:
        logger.info(f"Deleting the workspace-level group {display_name} with id {group_id}")
        self._ws.groups.delete(id=group_id)
        logger.info(f"Workspace-level group {display_name} with id {group_id} was deleted")
        return None  # should be deleted now
    except NotFound:
        return None  # it's definitely deleted now
```

Quite similar to how Terraform works

a) call RPC
b) set resource ID
*c) wait for **ready for use***

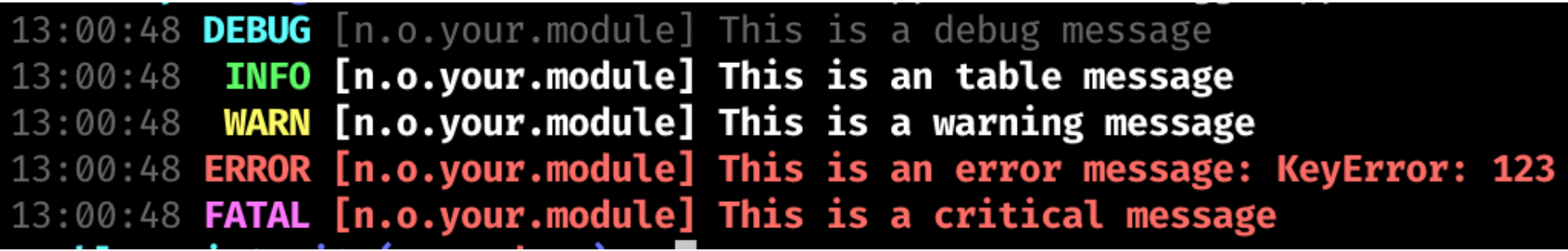
a) call backend by resource ID
b) set readonly fields
c) resolve documented eventual consistency

*a) compare **what is configured on platform** with **what we think is configured***
b) delete (or recreate) if READ by ID returns 404

… robust Python code starts to look like any code in GoLang, where we explicitly check for error returns for every single method call

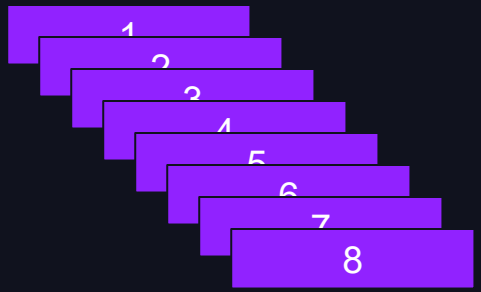
```
func (s *computeScanTask) Run(
    ctx context.Context) ([]alerts.Alert, error) {
    err := s.scanPools(ctx)
    if err != nil {
        return nil, err
    }
    allJobs, err := s.allJobs(ctx)
    if err != nil {
        return nil, err
    }
    err = s.scanClusters(ctx)
    if err != nil {
        return nil, err
    }
    err = s.scanClusterLibraries(ctx)
    if err != nil {
        return nil, err
    }
    err = s.scanJobs(ctx, allJobs)
    if err != nil {
        return nil, err
    }
    err = s.scanJobRuns(ctx)
    if err != nil {
        return nil, err
    }
    err = s.scanWarehouses(ctx)
    if err != nil {
        return nil, err
    }
    return s.alerts, nil
}
```

… most of our time is spent operating applications in production rather than constructing the code itself

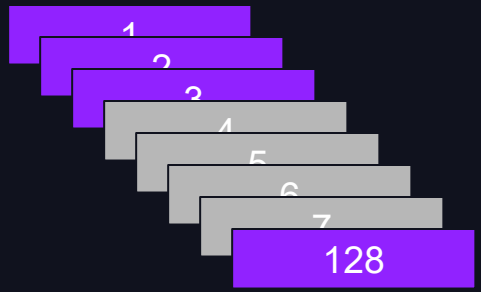


TYPICAL LOG-INTENSIVE WORKFLOW

By default, we only see the beginning and the end of standard output



Minute 1



Minute 5

1.5 hours

SOLUTION: LOG EVERY MINUTE TO WSFS

```
log_path = config_path.parent / "logs" / current_task.workflow / f"run-{workflow_run_id}"
log_path.mkdir(parents=True, exist_ok=True)

log_file = log_path / f"{task_name}.log"

...
file_handler = TimedRotatingFileHandler(log_file.as_posix(), when="M", interval=10)
log_format = "%(asctime)s %(levelname)s [%(name)s] {%(threadName)s} %(message)s"
log_formatter = logging.Formatter(fmt=log_format, datefmt="%H:%M:%S")
file_handler.setFormatter(log_formatter)
file_handler.setLevel(logging.DEBUG)

console_handler = _install(cfg.log_level)
databricks_logger.removeHandler(console_handler)
databricks_logger.addHandler(file_handler)
```

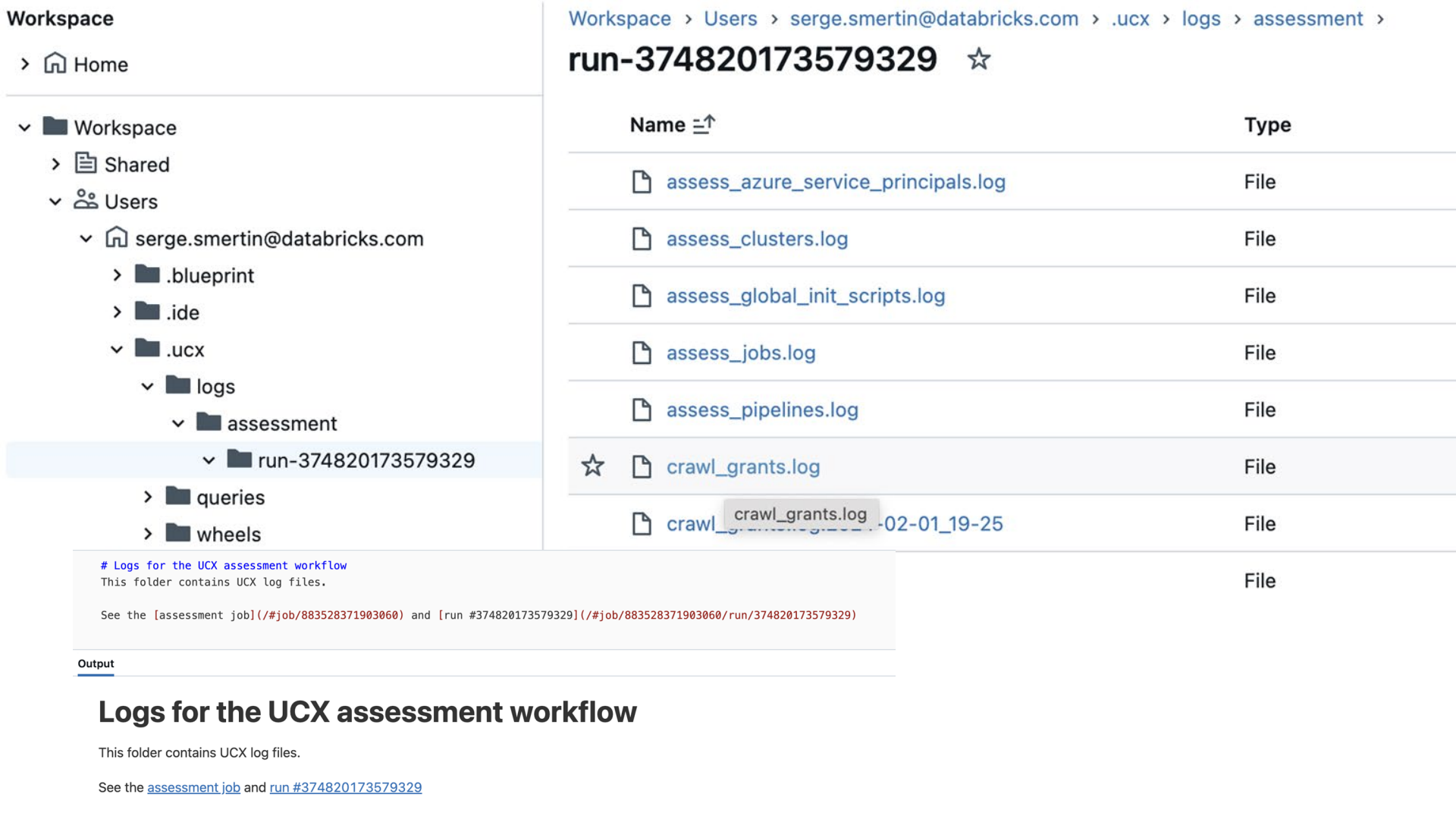
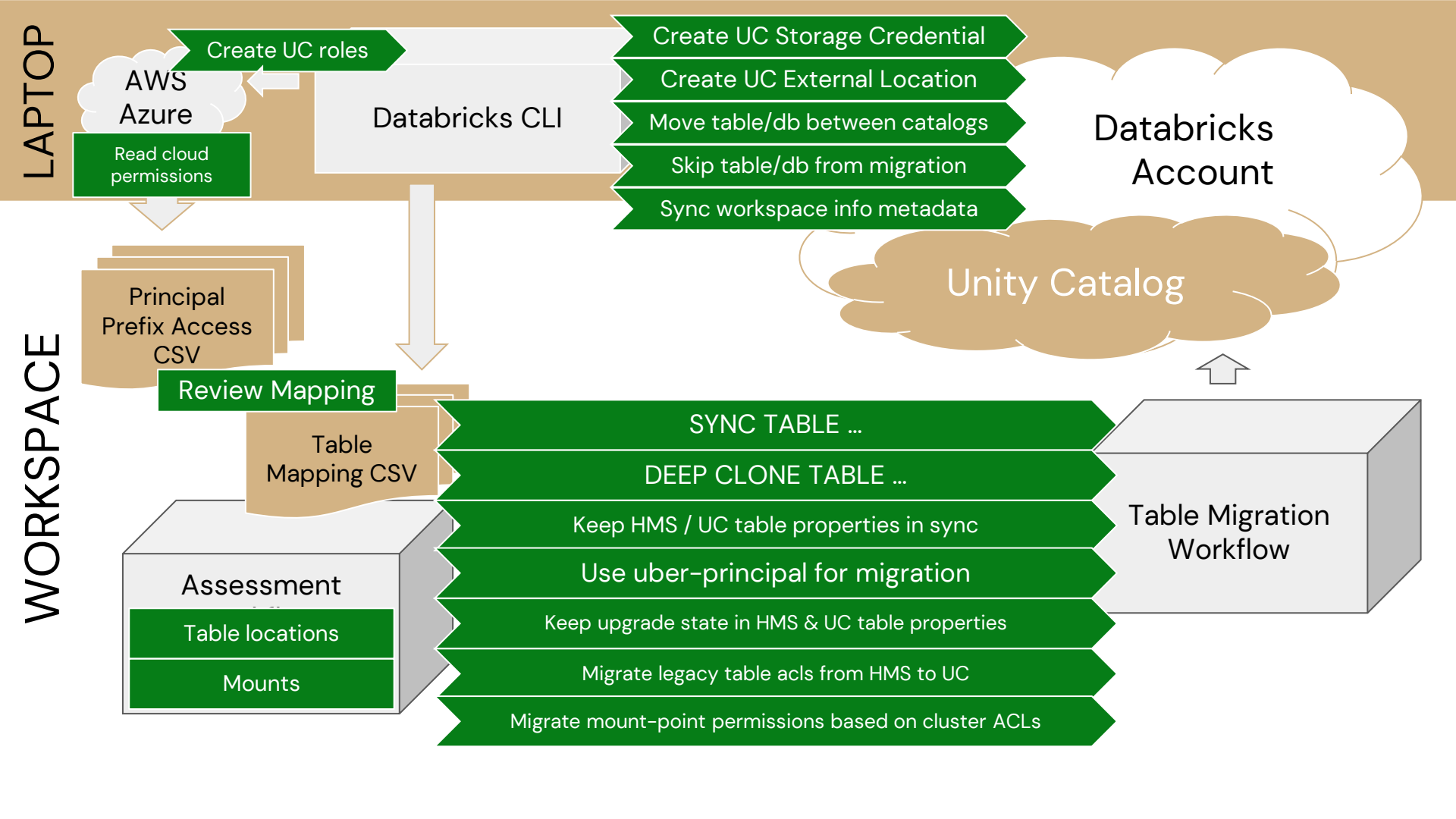
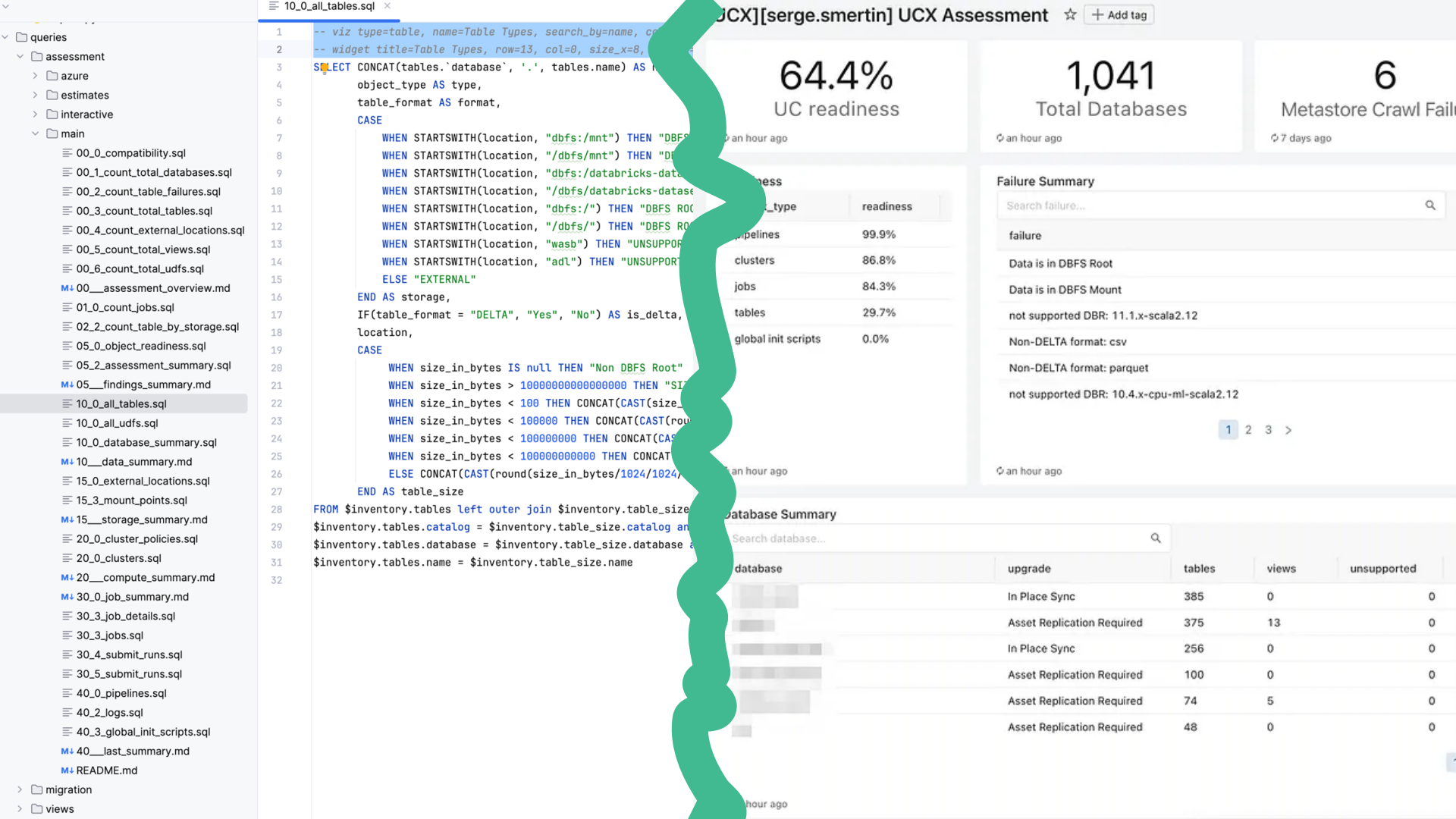

Workspace
Home
Workspace
Shared
Users
serge.smertin@databricks.com
.blueprint
.ide
.ucx
logs
assessment
run-374820173579329
queries
wheels
Workspace › Users › serge.smertin@databricks.com › .ucx › logs › assessment ›
run-374820173579329
Name
Type
assess_azure_service_principals.log
File
assess_clusters.log
File
assess_global_init_scripts.log
File
assess_jobs.log
File
assess_pipelines.log
File
crawl_grants.log
File
crawl_grants.log
File
Logs for the UCX assessment workflow
This folder contains UCX log files.
See the [assessment job](/#job/883528371903060) and [run #374820173579329](/#job/883528371903060/run/374820173579329)
Output
Logs for the UCX assessment workflow
This folder contains UCX log files.
See the assessment job and run #374820173579329

TABLE MIGRATION


LAPTOP
Create UC roles
AWS
Azure
Read cloud permissions
Databricks CLI
Create UC Storage Credential
Create UC External Location
Move table/db between catalogs
Skip table/db from migration
Sync workspace info metadata
Databricks Account
Unity Catalog
WORKSPACE
Principal Prefix Access CSV
Review Mapping
Table Mapping CSV
Assessment
Table locations
Mounts
SYNC TABLE ...
DEEP CLONE TABLE ...
Keep HMS / UC table properties in sync
Use uber-principal for migration
Keep upgrade state in HMS & UC table properties
Migrate legacy table acls from HMS to UC
Migrate mount-point permissions based on cluster ACLs
Table Migration Workflow

CHALLENGES:
-DASHBOARDS
-INTEGRATION TESTS
-UNRELEASED WHEELS


queries
assessment
azure
estimates
interactive
main
00_0_compatibility.sql
00_1_count_total_databases.sql
00_2_count_table_failures.sql
00_3_count_total_tables.sql
00_4_count_external_locations.sql
00_5_count_total_views.sql
00_6_count_total_udfs.sql
00___assessment_overview.md
01_0_count_jobs.sql
02_2_count_table_by_storage.sql
05_0_object_readiness.sql
05_2_assessment_summary.sql
05___findings_summary.md
10_0_all_tables.sql
10_0_all_udfs.sql
10_0_database_summary.sql
10___data_summary.md
15_0_external_locations.sql
15_3_mount_points.sql
15___storage_summary.md
20_0_cluster_policies.sql
20_0_clusters.sql
20___compute_summary.md
30_0_job_summary.md
30_3_job_details.sql
30_3_jobs.sql
30_4_submit_runs.sql
30_5_submit_runs.sql
40_0_pipelines.sql
40_2_logs.sql
40_3_global_init_scripts.sql
40___last_summary.md
README.md
migration
views
10_0_all_tables.sql
-- viz type=table, name=Table Types, search_by=name, co
-- widget title=Table Types, row=13, col=0, size_x=8,
SELECT CONCAT(tables.`database`, '.', tables.name) AS
object_type AS type,
table_format AS format,
CASE
WHEN STARTSWITH(location, "dbfs:/mnt") THEN "DBFS
WHEN STARTSWITH(location, "/dbfs/mnt") THEN "D
WHEN STARTSWITH(location, "dbfs:/databricks-dat
WHEN STARTSWITH(location, "/dbfs/databricks-datase
WHEN STARTSWITH(location, "dbfs:/") THEN "DBFS ROO
WHEN STARTSWITH(location, "/dbfs/") THEN "DBFS RO
WHEN STARTSWITH(location, "wasb") THEN "UNSUPPOR
WHEN STARTSWITH(location, "adl") THEN "UNSUPPORT
ELSE "EXTERNAL"
END AS storage,
IF(table_format = "DELTA", "Yes", "No") AS is_delta,
location,
CASE
WHEN size_in_bytes IS null THEN "Non DBFS Root"
WHEN size_in_bytes > 10000000000000000 THEN "SI
WHEN size_in_bytes < 100 THEN CONCAT(CAST(size_
WHEN size_in_bytes < 100000 THEN CONCAT(CAST(rou
WHEN size_in_bytes < 100000000 THEN CONCAT(CA
WHEN size_in_bytes < 100000000000 THEN CONCAT
ELSE CONCAT(CAST(round(size_in_bytes/1024/1024/
END AS table_size
FROM $inventory.tables left outer join $inventory.table_size
$inventory.tables.catalog = $inventory.table_size.catalog an
$inventory.tables.database = $inventory.table_size.database
$inventory.tables.name = $inventory.table_size.name
UCX][serge.smertin] UCX Assessment
Add tag
64.4%
UC readiness
an hour ago
1,041
Total Databases
an hour ness
_type
readiness
pelines
99.9%
clusters
86.8%
jobs
84.3%
tables
29.7%
global init scripts
0.0%
an hour ago
6
Metastore Crawl Fail
7 days ago
Failure Summary
Search failure...
failure
Data is in DBFS Root
Data is in DBFS Mount
not supported DBR: 11.1.x-scala2.12
Non-DELTA format: csv
Non-DELTA format: parquet
not supported DBR: 10.4.x-cpu-ml-scala2.12
1 2 3
an hour ago
atabase Summary
Search database...
database
upgrade
tables
views
unsupported
In Place Sync 385 0 0
Asset Replication Required 375 13 0
In Place Sync 256 0 0
Asset Replication Required 100 0 0
Asset Replication Required 74 5 0
Asset Replication Required 48 0 0
hour ago

```
def test_skip_unsupported_location(caplog):
    # mock crawled HMS external locations with two unsupported locations adl and wasbs
    ws = create_autospec(WorkspaceClient)
    mock_backend = MockBackend(
        rows={
            r"SELECT \* FROM location_test.external_locations": EXTERNAL_LOCATIONS[
                ("abfss://container1@test.dfs.core.windows.net/one/", 1),
                ("adl://container2@test.dfs.core.windows.net/", 2),
                ("wasbs://container2@test.dfs.core.windows.net/", 2),
            ]
        }
    )

    # mock listing UC external locations, no HMS external location will be matched
    ws.external_locations.list.return_value = [ExternalLocationInfo(name="none", url="none")]

    location_migration = location_migration_for_test(ws, mock_backend, mock_installation)
    location_migration.run()

    ws.external_locations.create.assert_called_once_with(
        "container1_test_one",
        "abfss://container1@test.dfs.core.windows.net/one/",
        "credential_sp1",
        comment="Created by UCX",
        read_only=False,
        skip_validation=False,
    )
    assert "Skip unsupported location: adl://container2@test.dfs.core.windows.net" in caplog.text
```

DATABRICKS LABS LSQL: LIGHTWEIGHT SQL ABSTRACTIONS

FINDING HIDDEN BUGS WITH PYLINT

(... and mypy, and ruff, …)

```
from airflow.providers.databricks.operators.databricks import
DatabricksCreateJobsOperator

tasks = [
    {
        "task_key": "banana",
        "notebook_task": {
            "notebook_path": "/Shared/test",
        },
        'new_cluster': {
            # [missing-data-security-mode] banana cluster missing `data_security_mode`
            # required for Unity Catalog compatibility
            "spark_version": "7.3.x-scala2.12",
            "node_type_id": "i3.xlarge",
            "num_workers": 2,
        },
    },
]

DatabricksCreateJobsOperator( #@
    task_id="jobs_create_named",
    tasks=tasks
)
```

databricks-airflow checker
W8901: missing-data-security-mode
W8902: unsupported-runtime

databricks-dbutils checker
R8903: dbutils-fs-cp
R8904: dbutils-fs-head
R8905: dbutils-fs-ls
R8906: dbutils-fs-mount
R8907: dbutils-credentials
R8908: dbutils-notebook-run
R8909: pat-token-leaked
R8910: internal-api

databricks-legacy checker
R8911: legacy-cli
W8912: incompatible-with-uc

databricks-notebooks checker
C8913: notebooks-too-many-cells
R8914: notebooks-percent-run

spark checker
C8915: spark-outside-function
C8917: use-display-instead-of-show
W8916: no-spark-argument-in-function

mocking checker
R8918: explicit-dependency-required
R8919: obscure-mock
R8921: mock-no-assign
R8922: mock-no-usage

eradicate checker
C8920: dead-code

WHY NOT (JUST) RUFF?

Even though RUFF is 10x+ faster than PyLint, it doesn't have a plugin system yet, nor does it have a feature parity with PyLint yet. Other projects use MyPy, Ruff, and PyLint together to achieve the most comprehensive code analysis. You can try using Ruff and just the checkers from this plugin in the same CI pipeline and pre-commit hook.

PYLINT PLUGIN FOR DATABRICKS

INTEGRATION TESTING

Run all tests in parallel

Retry individual tests until a timeout (in parallel)

Anti-flake: Try running failures one-by-one sequentially

Analyse test result runtime trend over time

```
08:04  INFO ✅test_some_entitlements (4.766s)
08:04  INFO ✅test_verify_permissions_for_redash (9.354s)
08:04  INFO 🦫 test_create_users (0.009s)
08:04  INFO ✅test_permission_for_files_anonymous_func (15.352s)
08:05  INFO ✅test_permission_for_udfs (41.028s)
08:05  INFO ✅test_job_cluster_policy (275.831s)
08:05  INFO ✅test_uninstallation (265.802s)
08:05  INFO ✅test_hms2hms_owner_permissions (67.206s)
08:05  INFO ✅test_instance_pools (5.678s)
08:05  INFO ✅test_verify_permission_for_udfs (5.264s)
08:05  INFO ✅test_clusters (6.440s)
08:06  INFO ✅test_replace_workspace_groups_with_account_groups (165.193s)
08:06  INFO ✅test_verify_permissions_for_secrets (1.941s)
08:07  INFO ✅test_job_failure_propagates_correct_error_message_and_logs (4
08:07  INFO ✅test_repair_run_workflow_job (333.909s)
08:08  INFO ✅test_delete_ws_groups_should_not_delete_current_ws_groups (1
08:08  INFO ✅test_running_real_validate_groups_permissions_job_fails (84.4
08:09  INFO ✅test_running_real_migrate_groups_job (537.850s)
08:10  INFO ✅test_running_real_assessment_job (577.490s)
08:10  INFO ✅test_running_real_validate_groups_permissions_job (99.348s)
08:11  INFO ✅test_running_real_remove_backup_groups_job (94.297s)
ERROR: failed: ❌99/100 passed, 1 failed, 7 skipped
```

Test failure:

`test_running_real_validate_groups_permissi`

Open github-actions bot opened this issue on Apr 11 · 5 comments

github-actions bot commented on Apr 11

▸ ❌ test_running_real_validate_groups_permissions_job_fails: databricks.labs.blueprint.parallel.ManyE
failures: Unknown: parse_logs: RuntimeError: Found error log records: (2m31.861s)

Running from nightly #24

github-actions bot added the bug label on Apr 11

github-actions bot commented on Apr 17

▾ ❌ test_running_real_validate_groups_permissions_job_fails: TimeoutError: Timed out after 0:05:00

```
TimeoutError: Timed out after 0:05:00
[gw9] linux -- Python 3.10.14 /home/runner/work/ucx/ucx/.venv/bin/python
07:06 DEBUG [databricks.labs.ucx.mixins.fixtures] added workspace user fixture: User(active=True, display_name
07:06 INFO [databricks.labs.ucx.mixins.fixtures] Workspace group ucx_s1UQ: https://DATABRICKS_HOST#setting/acc
07:06 DEBUG [databricks.labs.ucx.mixins.fixtures] added workspace group fixture: Group(display_name='ucx_s1UQ'
07:06 INFO [databricks.labs.ucx.mixins.fixtures] Account group ucx_s1UQ: https://accounts.CLOUD_ENVdatabricks.
07:06 DEBUG [databricks.labs.ucx.mixins.fixtures] added account group fixture: Group(display_name='ucx_s1UQ',
07:06 INFO [databricks.labs.ucx.mixins.fixtures] Cluster policy: https://DATABRICKS_HOST#setting/clusters/clus
07:06 DEBUG [databricks.labs.ucx.mixins.fixtures] added cluster policy fixture: CreatePolicyResponse(policy_id
07:06 DEBUG [databricks.labs.ucx.mixins.fixtures] added cluster_policy permissions fixture: 001DC1FFFF043CBE [
07:06 INFO [databricks.labs.ucx.mixins.fixtures] Schema hive_metastore.ucx_sfgty: https://DATABRICKS_HOST/expl
07:06 DEBUG [databricks.labs.ucx.mixins.fixtures] added schema fixture: SchemaInfo(browse_only=None, catalog_n
07:06 DEBUG [databricks.labs.ucx.install] Cannot find previous installation: Path (/Users/0a330eb5-dd51-4d97-b
07:06 INFO [databricks.labs.ucx.install] Please answer a couple of questions to configure Unity Catalog migrat
07:06 INFO [databricks.labs.ucx.installer.hms_lineage] HMS Lineage feature creates one system table named syst
07:06 INFO [databricks.labs.ucx.install] Fetching installations...
07:06 INFO [databricks.labs.ucx.installer.policy] Creating UCX cluster policy.
07:06 DEBUG [tests.integration.conftest] Waiting for clusters to start...
07:11 DEBUG [tests.integration.conftest] Waiting for clusters to start...
07:11 INFO [databricks.labs.ucx.install] Installing UCX v0.21.1+7720240417071157
07:11 INFO [databricks.labs.ucx.install] Creating ucx schemas...
07:12 INFO [databricks.labs.ucx.installer.workflows] Creating new job configuration for step=migrate-groups
07:12 INFO [databricks.labs.ucx.installer.workflows] Creating new job configuration for step=migrate-tables-in-
07:12 INFO [databricks.labs.ucx.installer.workflows] Creating new job configuration for step=remove-workspace-
07:12 INFO [databricks.labs.ucx.installer.workflows] Creating new job configuration for step=migrate-groups-ex
```

github-actions bot commented 1 hour ago • edited

✅ 173/173 passed, 1 flaky, 25 skipped, 2h26m26s total

Flaky tests:

- 🤪 test_running_real_migrate_groups_job (6m50.728s)

Running from acceptance #3462

```
================ 1 passed, 197 deselected in 369.63s (0:0
Warning: 🥴 flaky test detected: test_running_real_migrate
✅ 173/173 passed, 1 flaky, 25 skipped, 2h26m26s total, to
2024/05/23 16:28:46 Stopping tail as file no longer exists
test_running_real_migrate_groups_job.out
```

Lock conversation

Pin issue

Transfer issue

Convert to discussion

Delete issue

Integration testing with pytest

```python
@retried(on=[NotFound, Unknown, InvalidParameterValue], timeout=timedelta(minutes=20))
def test_running_real_assessment_job(
    ws, new_installation, make_ucx_group, make_cluster_policy,
make_cluster_policy_permissions
):
    ws_group_a, acc_group_a = make_ucx_group()

    cluster_policy = make_cluster_policy()
    make_cluster_policy_permissions(
        object_id=cluster_policy.policy_id,
        permission_level=PermissionLevel.CAN_USE,
        group_name=ws_group_a.display_name,
    )

    install = new_installation(lambda wc: replace(wc,
                               include_group_names=[ws_group_a.display_name]))
    install.run_workflow("assessment")

    generic_permissions = GenericPermissionsSupport(ws, [])
    after = generic_permissions.load_as_dict("cluster-policies", cluster_policy.policy_id)
    assert after[ws_group_a.display_name] == PermissionLevel.CAN_USE
```

DEBUGGING TESTS

```python
def test_this_wheel_installs(ws, wsfs_wheel):
    commands = CommandExecutor(ws)

    commands.install_notebook_library(f"/Workspace{wsfs_wheel}")
    installed_version = commands.run(
        """
        from databricks.labs.ucx.__about__ import __version__
        print(__version__)
        """
    )

    assert installed_version is not None


def test_sql_backend_works(ws, wsfs_wheel):
    commands = CommandExecutor(ws)

    commands.install_notebook_library(f"/Workspace{wsfs_wheel}")
    database_names = commands.run(
        """
        from databricks.labs.ucx.tacl._internal import RuntimeBackend
        backend = RuntimeBackend()
        return backend.fetch("SHOW DATABASES")
        """
    )

    assert len(database_names) > 0
```

test_this_wheel_installs()

Git | Debug | Commit | Python Packages | TODO | Problems | Terminal | Python Console | Services

Tests passed: 0 (a minute ago)

```
default_cluster_id = env_or_skip("TEST_DEFAULT_CLUSTER_ID")
tacl_cluster_id = env_or_skip("TEST_LEGACY_TABLE_ACL_CLUSTER_ID")
```

```
@pytest.fixture
def env_or_skip(debug_env) -> Callable[[str], str]:
    skip = pytest.skip
    if _is_in_debug():
        skip = pytest.fail  # typ

    Serge Smertin
    def inner(var: str) -> str:
        if var not in debug_env:
            skip(f"Environment variable {var} is missing")
        return debug_env[var]

    return inner
```

2 usages Serge Smertin

```
def _is_in_debug() -> bool:
    return os.path.basename(sys.argv[0]) in {"_jb_pytest_runner.py", "testlauncher.py"}
```

```
@pytest.fixture
def debug_env(monkeypatch, debug_env_name) -> MutableMapping[str, str]:
    if not _is_in_debug():
        return os.environ
    conf_file = pathlib.Path.home() / ".databricks/debug-env.json"
    if not conf_file.exists():
        return os.environ
    with conf_file.open("r") as f:
        conf = json.load(f)
        if debug_env_name not in conf:
            sys.stderr.write(
                f"""..."""
            )
            msg = f"{debug_env_name} not found in ~/.databricks/debug-env.json"
            raise KeyError(msg)
        for k, v in conf[debug_env_name].items():
            monkeypatch.setenv(k, v)
    return os.environ
```

SAME APPROACH IS USED FOR MULTI-CLOUD TESTING OF …

- Databricks Terraform Provider
- Databricks CLI
- Databricks VS Code extension
- Databricks SDK for Python, Go, Java
- Databricks Labs UCX
- Databricks Labs Watchdog

“static fixtures” are created by Terraform

“dynamic fixtures” that require a cleanup

Very rough equivalent of something between "Chaos Monkey" and "Cloud Custodian"

"dynamic fixtures" that require a cleanup

DATABRICKS LABS BLUEPRINT

DEPLOY WHEELS

```
import logging

from databricks.sdk import WorkspaceClient
from databricks.labs.blueprint.wheels import ProductInfo

logger = logging.getLogger(__name__)

product_info = ProductInfo(__file__)

version = product_info.unreleased_version()
is_git = product_info.is_git_checkout()
is_unreleased = product_info.is_unreleased_version()

logger.info(f'Version is: {version}')
logger.info(f'Git checkout: {is_git}')
logger.info(f'Is unreleased: {is_unreleased}')

w = WorkspaceClient()
installation = product_info.current_installation(w)
with product_info.wheels(w) as wheels:
    remote_wheel = wheels.upload_to_wsfs()
    logger.info(f'Uploaded to {remote_wheel}')
    wheel_paths = wheels.upload_wheel_dependencies(...)
    for path in wheel_paths:
        print(f'Uploaded dependency to {path}')
```

CONFIG EVOLUTION

```python
from databricks.labs.blueprint.installation import Installation

@dataclass
class EvolvedConfig:
    __file__ = "config.yml"
    __version__ = 3

    initial: int
    added_in_v1: int
    added_in_v2: int

    @staticmethod
    def v1_migrate(raw: dict) -> dict:
        raw["added_in_v1"] = 111
        raw["version"] = 2
        return raw

    @staticmethod
    def v2_migrate(raw: dict) -> dict:
        raw["added_in_v2"] = 222
        raw["version"] = 3
        return raw

installation = Installation.current(WorkspaceClient(),
  "blueprint")
cfg = installation.load(EvolvedConfig)

assert 999 == cfg.initial
assert 111 == cfg.added_in_v1  # <-- added by v1_migrate()
assert 222 == cfg.added_in_v2  # <-- added by v2_migrate()
```

APP (OR DATABASE) EVOLUTION


```python
from ... import Config
from databricks.sdk import WorkspaceClient
from databricks.labs.blueprint.upgrades import Upgrades
from databricks.labs.blueprint.wheels import ProductInfo

product_info = ProductInfo.from_class(Config)
ws = WorkspaceClient(product=product_info.product_name(),
product_version=product_info.version())
installation = product_info.current_installation(ws)
config = installation.load(Config)
upgrades = Upgrades(product_info, installation)
upgrades.apply(ws)
```

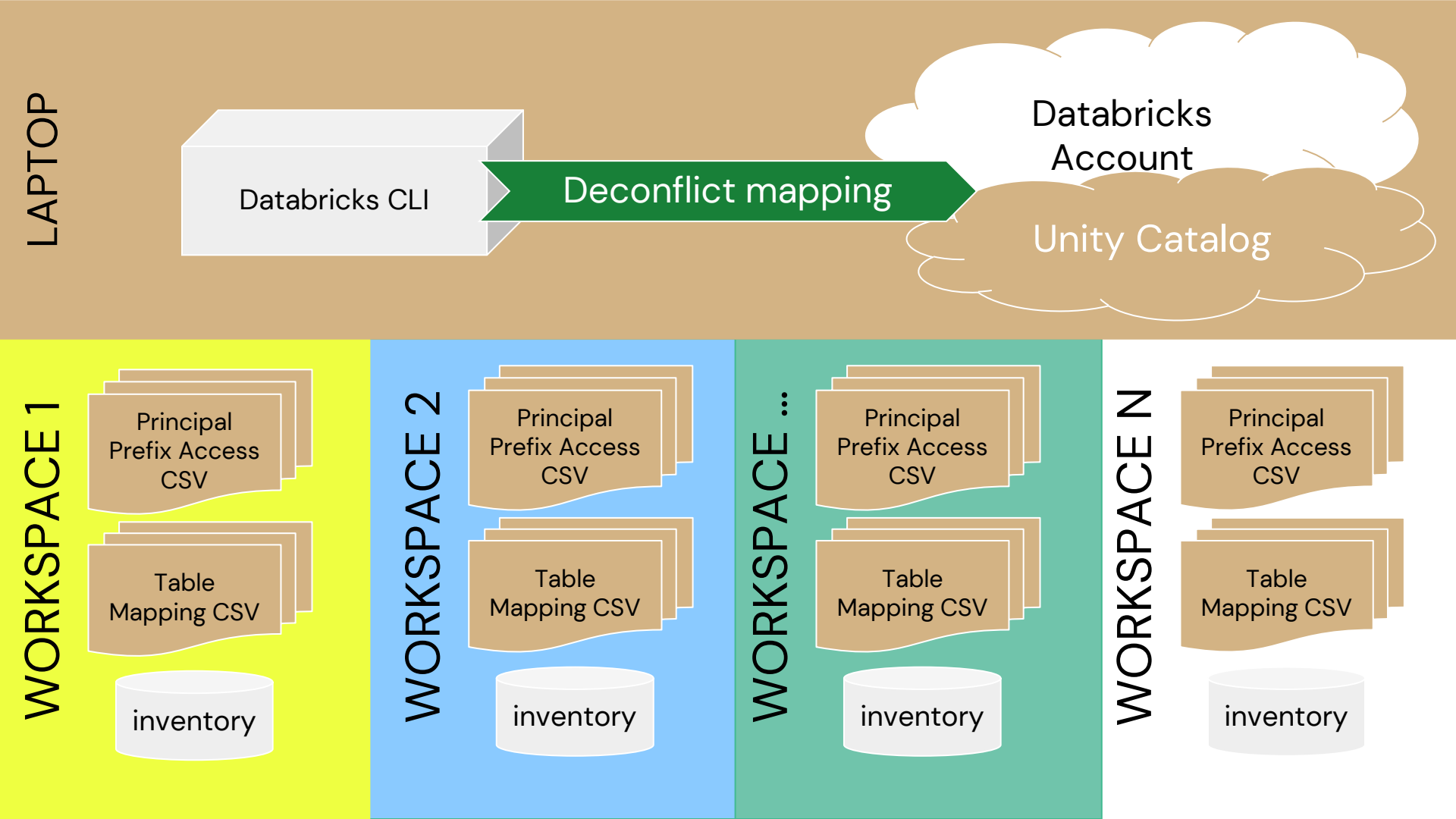
```python
# and in v0.0.1_add_service.py:

from ... import Config

import logging, dataclasses
from databricks.sdk import WorkspaceClient
from databricks.labs.blueprint.installation import Installation

upgrade_logger = logging.getLogger(__name__)

def upgrade(installation: Installation, ws: WorkspaceClient):
    upgrade_logger.info("creating new automated service user")
    config = installation.load(Config)
    service_principal = ws.service_principals.create(
      display_name='blueprint-service')
    new_config = dataclasses.replace(config,
      application_id=service_principal.application_id)
    installation.save(new_config)
```

PARALLEL THINGS

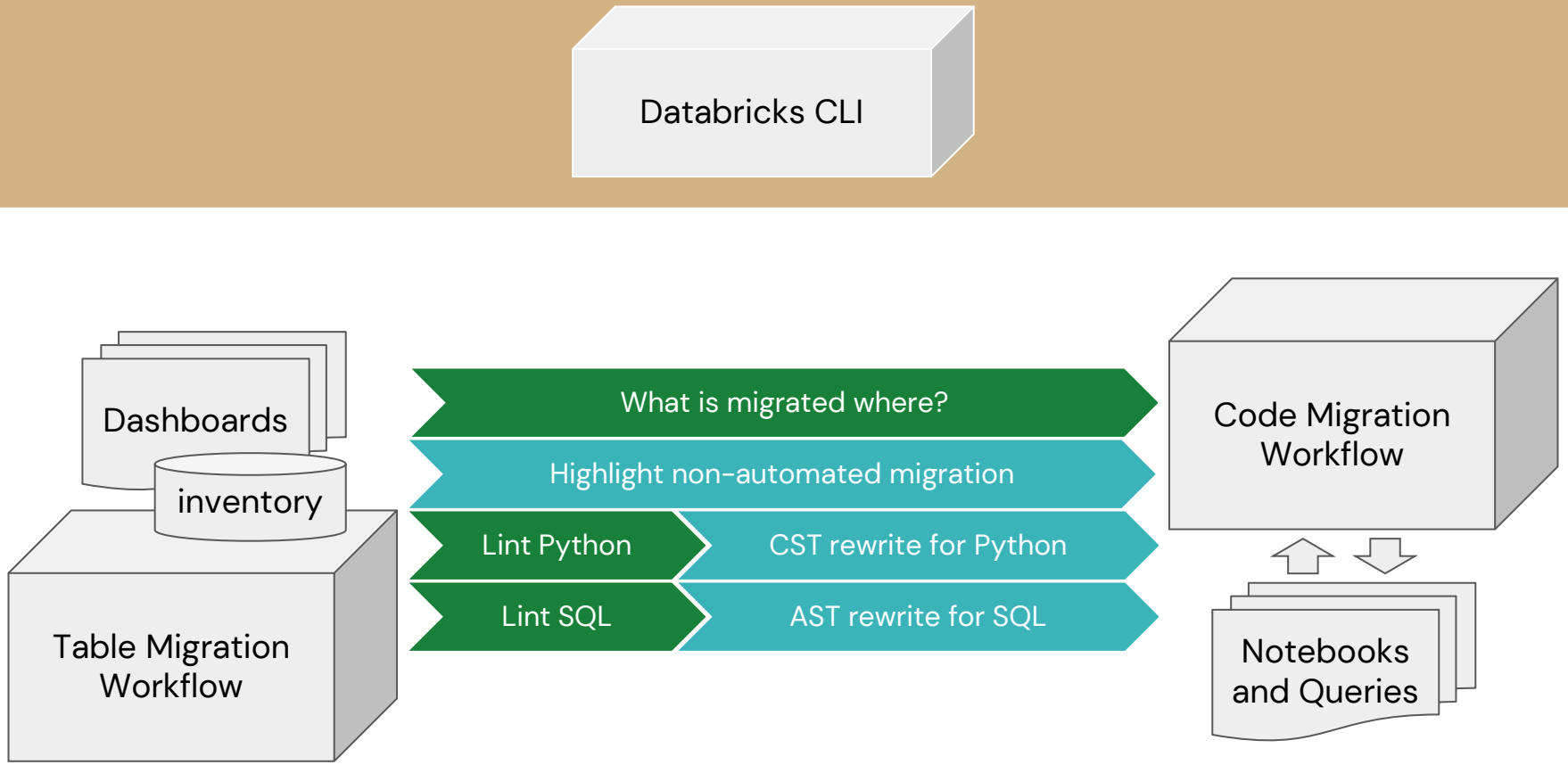
```
from databricks.sdk.errors import NotFound
from databricks.labs.blueprint.parallel import Threads

def works():
   return True

def fails():
   raise NotFound("something is not right")

tasks = [works, fails, works, fails, works, fails,
         works, fails]
results, errors = Threads.gather("doing some work", tasks)

assert [True, True, True, True] == results
assert 4 == len(errors)
```

```
14:08:31 ERROR [d.l.blueprint.parallel][doing_some_work_0]
doing some work task failed: something is not right: ...
...
14:08:31 ERROR [d.l.blueprint.parallel][doing_some_work_3]
doing some work task failed: something is not right: ...
14:08:31 ERROR [d.l.blueprint.parallel] More than half
'doing some work' tasks failed: 50% results available
(4/8). Took 0:00:00.001011
```

DATABRICKS LABS BLUEPRINT: PRODUCTION-FOCUSED UTILITIES


LAPTOP
Databricks CLI
Deconflict mapping
Databricks Account
Unity Catalog
WORKSPACE 1
Principal Prefix Access CSV
Table Mapping CSV
inventory
WORKSPACE 2
Principal Prefix Access CSV
Table Mapping CSV
inventory
WORKSPACE ...
Principal Prefix Access CSV
Table Mapping CSV
inventory
WORKSPACE N
Principal Prefix Access CSV
Table Mapping CSV
inventory

CODE MIGRATION


LAPTOP
Databricks CLI
Dashboards
inventory
Table Migration Workflow
What is migrated where?
Highlight non-automated migration
Lint Python
CST rewrite for Python
Lint SQL
AST rewrite for SQL
Code Migration Workflow
Notebooks and Queries

THANK YOU

linkedin.com/in/smertin
github.com/nfx
ssmertin.com