

```
# MZ2POL-07: Validation and Troubleshooting

> **Series:** MZ2POL | **Notebook:** 8 of 8 | **Created:** December 2025

## Overview

This notebook provides comprehensive validation procedures and troubleshooting guidance for your migration from Management Zones to Policies, Boundaries, and Segments. Use this guide to verify your migration success and resolve common issues.

## Prerequisites



- Completed migration (MZ2POL-06)
- Admin access to Dynatrace
- Test user accounts for validation



## Learning Objectives

By the end of this notebook, you will:


1. Be able to validate migration completeness
2. Know how to troubleshoot common access issues
3. Understand ongoing maintenance requirements
4. Have queries for monitoring access health



---

## 1. Migration Validation Checklist

### Component Validation

| Component | Validation | Status |
|-----|-----|-----|
| User Groups | All groups created and populated |  |
| Policies | Policies bound to all groups |  |
| Boundaries | Boundaries applied where needed |  |
| Security Context | All entities have security context |  |
| Segments | Segments created for all filtering needs |  |
| Dashboards | Updated to use segments |  |
| Alerts | Updated (when segment support available) |  |

### Access Validation Matrix

For each user type, verify:

| User Type | Can Access | Cannot Access | Verified |
|-----|-----|-----|-----|
```

```
| Frontend Dev | Frontend services | Backend services | □ |
| SRE | All production | Development | □ |
| Viewer | View dashboards | Edit settings | □ |

---

## 2. Security Context Validation

### Check Coverage

```dql
// Validate: All services have security context
fetch dt.entity.service
| summarize
 total = count(),
 withContext = countIf(isNotNull(dt.security_context)),
 missingContext = countIf(isNull(dt.security_context))
| fields total, withContext, missingContext,
 coveragePercent = round(100.0 * withContext / total, decimals: 2)
```

```dql
// Find services missing security context
// These need to be addressed for complete migration
fetch dt.entity.service
| filterisNull(dt.security_context)
| fields entity.name, tags, managementZones
| sort entity.name asc
| limit 50
```

```dql
// Validate: Host security context coverage
fetch dt.entity.host
| summarize
 total = count(),
 withContext = countIf(isNotNull(dt.security_context)),
 missingContext = countIf(isNull(dt.security_context))
| fields total, withContext, missingContext,
 coveragePercent = round(100.0 * withContext / total, decimals: 2)
```

### Verify Security Context Distribution

```dql
// Distribution of security context values
// Verify expected teams/environments are represented
fetch dt.entity.service
```

```
| filter isNotNull(dt.security_context)
| summarize count = count(), by:{dt.security_context}
| sort count desc
```
```
3. Access Control Validation

Test User Access Scenarios

For each test scenario:
1. Log in as test user
2. Navigate to target app/data
3. Verify expected access
4. Document results

Scenario 1: Team-Based Access

Test User: Member of `frontend-team` group
Expected: Access to frontend services only

Validation Steps:
1. Open Services app
2. Select "Frontend Team" segment
3. Verify only frontend services visible
4. Try accessing backend service directly → Should fail

Scenario 2: Environment Restriction

Test User: Member of `prod-viewers` group
Expected: View production data only, no edit

Validation Steps:
1. Open Logs app
2. Select "Production" segment
3. Query logs → Should succeed
4. Try editing settings → Should fail
5. Select different segment → Should show no data (if boundary restricts)

Scenario 3: Full Access (SRE)

Test User: Member of `sre-team` group
Expected: Full access to production environment

Validation Steps:
1. Open any app
2. Verify data visible across all services
```

```

3. Verify edit capabilities work
4. Verify settings access

4. Segment Validation

Verify Segment Data Match

Compare segment results with expected MZ results:

```dql
// Compare: MZ entity count vs Segment entity count
// Replace with your MZ name and segment filter
fetch dt.entity.service
| summarize
    viaMZ = countIf(in(managementZones, {"Frontend-Team"})),
    viaSegment = countIf(matchesValue(tags, "team:frontend"))
| fields viaMZ, viaSegment,
    match = if(viaMZ == viaSegment, then: "MATCH", else: "MISMATCH")
```

```dql
// Find entities in MZ but not matching segment filter
// These are migration gaps
fetch dt.entity.service
| filter in(managementZones, {"Frontend-Team"})
| filter NOT matchesValue(tags, "team:frontend")
| fields entity.name, tags, managementZones
```

```dql
// Find entities matching segment but not in MZ
// May indicate over-inclusion in new model
fetch dt.entity.service
| filter matchesValue(tags, "team:frontend")
| filter NOT in(managementZones, {"Frontend-Team"})
| fields entity.name, tags, managementZones
```

5. Common Issues and Troubleshooting

![Troubleshooting Flow]
()

```

b3AgB2Zmc2V0PSIgC3R5bGU9InN0b3AtY29sb3I6I2VmNDQ0NDtzdG9wLW9wYWNPdHk6MSlGz4KICAgICAgnPHN0b3Agb2Zmc2V0PSIxMDAIIiBzdHlsZT0ic3RvcC1jb2xvcjojZGMyNjI203N0b3Atb3BhY2l0eToxiAvPgogICAgnPC9saW5LYXJHcmFkaWVudD4KICAgIDxsaw5LYXJHcmFkaWVudCBpZD0iY2h1Y2tHcmFkiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAIIj4KICAgICAgnPHN0b3Agb2Zmc2V0PSIxJSIgc3R5bGU9InN0b3AtY29sb3I6IzNiODJmNjtzdG9wLW9wYWNPdHk6MSlGz4KICAgICAgnPHN0b3Agb2Zmc2V0PSIxMDAIIiBzdHlsZT0ic3RvcC1jb2xvcjojMju2M2Vi03N0b3Atb3BhY2l0eToxiAvPgogICAgnPC9saW5LYXJHcmFkaWVudD4KICAgIDxsaw5LYXJHcmFkaWVudCBpZD0iZml4R3jhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAIIiB5Mj0iMTAwJSI+CiAgICAgnIDxdG9wIG9mZnNldD0iMCUiIHn0eWx1PSJzdG9wLWNVbG9y0iMyMmM1NWU7c3RvcC1vcGFjaXR50jEiIC8+CiAgICAgnIDxdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzE2YTM0YTtzdG9wLW9wYWNPdHk6MSlGz4KICAgIDwvbgLuzWfYr3JhZGllbnQ+CiAgICA8ZmlsdGVyIGlkPSJ0c1NoYWRvdyI+CiAgICAgnIDxmZURyb3BTaGFkb3cgZHg9IjEiIGR5PSIxIiBzdGREZXZpYXRpb249IjIiIGZsb29kLW9wYWNPdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogICAgnPG1hcmtlcibpZD0idHNbcnjvdIgBWfya2VyV2lkGg9IjEwIiBtYXJrZXJIZWlnaHQ9IjciIhjlzlg9IjkIiHjlzlk9IjMuNSIgb3JpzW50PSJhdXRvIj4KICAgICAgnPHBvbHlnb24gcG9pbnRzPSIxIDAsIDEwIDMuNSwgMCA3IiBmaWxsPSIjNjQ3NDhiIi8+CiAgICA8L21hcmtlcj4KICA8L2R1ZnM+CgogIDwhLS0gQmFja2dyb3VuZCATLT4KICA8cmVjdCB3aWR0aD0i0DAwIiBoZWlnaHQ9IjM4MCigZmlsbD0iI2Y4ZjlmYSIgcng9IjEwIi8+CgogIDwhLS0gVG10bGUqLS0+CiAgPHRleHQgeD0iNDAwIiB5PSIyOCigZm9udC1mYW1pbHk9IkJFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE4IiBmb250LXdlaWdodD0iYm9sZC1gZmlsbD0iIzMzMyIgdGV4dC1hbmnob3I9Im1pZGRsZSI+QWNjZXnZENvbnRy2wgVHJvdWJsZXNb290aW5nIEZsb3c8L3R1eHQ+CgogIDwhLS0gSXNzdWUgMTogQ2FuJ3QgU2V1IERhdGEgLS0+CiAgPHJ1Y3QgeD0iMzAiIh9IjU1IiB3aWR0aD0iMjMwIiBoZWlnaHQ9IjE0NSIgcn9IjgiIGZpbGw9IiNmZmYiIHn0cm9rZT0iI2VmNDQ0NCIgc3Ryb2t1LXdPZHRoPSIyIiBmaWx0ZXi9InVybCgjdHNTaGFkb3cpIi8+CiAgPHJ1Y3QgeD0iMzAiIh9IjU1IiB3aWR0aD0iMjMwIiBoZWlnaHQ9IjMwIiByeD0iOCigZmlsbD0idXjsKCnpc3N1ZUdyYWQpIi8+CiAgPHRleHQgeD0iMTQ1IiB5PSI3NSIgZm9udC1mYW1pbHk9IkJFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGuIHRleHQtYW5jaG9yPSJtaWRkbGuipkNhbid0IFnlZSBbknkRGF0YTwdGV4d4KCIgPHRleHQgeD0iNDUiIh9IjEwMCigZm9udC1mYW1pbHk9IkJFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjMzIj5DaGVjazo8L3R1eHQ+CiAgPHRleHQgeD0iNDUiIh9IjExNSIgZm9udC1mYW1pbHk9IkJFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj4xLiBhcm91cCbtZw1iZXJzaGlwPC90Zxh0PgogIDx0Zxh0IHg9IjQ1IiB5PSIxMzAiIGZvbnQtZmFtaWx5PSJBcmhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCiGZmlsbD0iIzY0NzQ4YiI+Mi4gUG9saWN5IGJpbmRpBmcgZxhpc3RzPC90Zxh0PgogIDx0Zxh0IHg9IjQ1IiB5PSIxNDUiIIGZvbnQtZmFtaWx5PSJBcmhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCiGZmlsbD0iIzY0NzQ4YiI+My4gRGF0YSBwb2xpY3kgaW5jbhVkvZwQ8L3R1eHQ+CiAgPHRleHQgeD0iNDUiIh9IjE2MCigZm9udC1mYW1pbHk9IkJFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj40LiBCb3VuZGFyeSBub3QgdG9vIHJlc3RyaWn0axZlPC90Zxh0PgogIDx0Zxh0IHg9IjQ1IiB5PSIx0DIiIhdPZHRoPSIyMDAiIGHlaWdodD0iMTIiIhJ4PSIyIiBmaWxsPSIjZDFmYWU1Ii8+CiAgPHRleHQgeD0iMTQ1IiB5PSIx0TiIiGZvbnQtZmFtaWx5PSJBcmhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCiGZmlsbD0iIzY0NzQ4YiI+NS4gU2VnbWVudCBzzWx1Y3Rpb248L3R1eHQ+CgogIDxyZwn0IHg9IjQ1IiB5PSIx0DIiIhdPZHRoPSIyMDAiIGHlaWdodD0iMTIiIhJ4PSIyIiBmaWxsPSIjZDFmYWU1Ii8+CiAgPHRleHQgeD0iMTQ1IiB5PSIx0TiIiGZvbnQtZmFtaWx5PSJBcmhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCiGZmlsbD0iIzY0NzQ4YiI+NS4gU2VnbWVudCBzzWx1Y3Rpb248L3R1eHQ+CgogIDxyZwn0IHg9IjQ1IiB5PSIx0DIiIhdPZHRoPSIyMDAiIGHlaWdodD0iMTIiIhJ4PSIyIiBmaWxsPSIjZDFmYWU1Ii8+CiAgPHRleHQgeD0iMTQ1IiB5PSIx0TiIiGZvbnQtZmFtaWx5PSJBcmhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCiGZmlsbD0iIzA0Nzg1NyIgdGV4dC1hbmnob3I9Im1pZGRsZSI+Rml40iBBZGQgbWlzc2luZyBwb2xpY3kvYm91bmRhcnk8L3R1eHQ+CgogIDwhLS0gSXNzdWUgMjogU2VlcbyUb28gTXVjaCATLT4KICA8cmVjdCB4PSIy0DUiIh9IjU1IiB3aWR0aD0iMjMwIiBoZWlnaHQ9IjE0NSIgcn9IjgiIGZpbGw9IiNmZmYiIHn0cm9rZT0iI2Y10WUwYiIgc3Ryb2t1LXdPZHRoPSIyIiBmaWx0ZXi9InVybCgjdHNTaGFkb3cpIi8+CiAgPHJ1Y3QgeD0iMjg1IiB5PSI1NSIg21kdGg9IjIzMCigaGVpZ2h0PSIxMCiGcng9IjgiIGZpbGw9IiNmNTlLMGIiLz4KICA8dGV4dCB4PSI0MDAiIh9IjciIiBmb250LWzbWlseT0iQXJpYlwseIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2Vp

Z2h0PSJib2xkIiBmaWxsPSJ3aG10ZSIgdGV4dC1hbhNob3I9Im1pZGRsZSI+U2VlcyBUb28gTXVja  
CBEYXRhPC90ZXh0Pg0KICA8dGV4dCB4PSIzM0AiIHk9IjEwMCigZm9udC1mYW1pbHk9IkFyaWFsLC  
BzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjMzMzIj5DaGVjazo8L3RleHQ+CiAgPHR  
leHQgeD0iMzAwIiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJBcmhbCwg2Fucy1zZXJpZiIgZm9udC1z  
aXplPSIxMCigZmlsbD0iIzY0NzQ4YiI+MS4gTXVsdlwbGugZ3JvdXAzbWtYmVyc2hpcHM8L3Rle  
HQ+CiAgPHRleHQgeD0iMzAwIiB5PSIxMzAiiIGZvbnQtZmFtaWx5PSJBcmhbCwg2Fucy1zZXJpZi  
IgZm9udC1zaXplPSIxMCigZmlsbD0iIzY0NzQ4YiI+Mi4gQm91bmRhcnkgY29uZGl0aW9uczwvdGV  
4d4KICA8dGV4dCB4PSIzM0AiIHk9IjE0NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlm  
IiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj4zLiBTZWN1cml0eSBjb250ZXh0IG9uIGVud  
Gl0aWVzPC90ZXh0PgogIDx0ZXh0IHg9IjMwMCiGeT0iMTYwIiBmb250LWzbhWlseT0iQXJpYwvsiH  
NhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM2NDc00GIiPjQuIEFsbCAzIGRvbWFpbnM  
gaW4gYm91bmRhcnk8L3RleHQ+CiAgPHRleHQgeD0iMzAwIiB5PSIxNzUiIGZvbnQtZmFtaWx5PSJB  
cmhbCwg2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCigZmlsbD0iIzY0NzQ4YiI+NS4gUG9saW5I  
GNvbmRpdGlvbnM8L3RleHQ+CgogIDxyZWN0IHg9IjMwMCiGeT0iMTgyIiB3aWR0aD0iMjAwIiBoZW  
lnaHQ9IjEyiByeD0iMiIgZmlsbD0iI2ZlZjNjNyIvPgogIDx0ZXh0IHg9IjQwMCiGeT0iMTkyIiB  
mb250LWzbhWlseT0iQXJpYwvsiHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM5MjQw  
MGUiIHRleHQtYw5jaG9yPSJtaWrbGUipkZpeDogVGlnaHRlbiB1b3VuZGFyeSBjb25kaXRpb25zP  
C90ZXh0Pg0KICA8IS0tIELzc3VlIDM6IENsYXNzaWmgV29ya3MsIE5ldyEBe2Vzbid0IC0tPgogID  
xyZWN0IHg9IjU0MCiGeT0iNTUiIHdpZHRoPSIyMzAiiGhlaWdodD0iMTQ1IIiByeD0i0CIgZmlsbD0  
iI2ZmZiIgc3Ryb2t1PSIj0GI1Y2Y2IiBzdHJva2Utd2lkGg9IjIiIGZpbHrlcj0idXjsKCn0c1No  
YWRdykiLz4KICA8cmVjdCB4PSI1NDAiIHk9IjU1iB3aWR0aD0iMjMwIiBoZwnaHQ9IjMwIiBye  
D0i0CIgZmlsbD0iIzhinWNmNiIvPgogIDx0ZXh0IHg9IjY1NSIgeT0iNzUiIGZvbnQtZmFtaWx5PS  
JBcmhbCwg2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZwnaHQ9IjvGQiIGZpbGw  
9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZgxlij5DbGFzc2ljIE9LLCB0ZXcgQXBwcyBGywlsPC90  
ZXh0Pg0KICA8dGV4dCB4PSI1NTUiIHk9IjEwMCigZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlc  
mlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjMzMzIj5DaGVjazo8L3RleHQ+CiAgPHRleHQgeD0iNT  
U1IiB5PSIxMTUiIGZvbnQtZmFtaWx5PSJBcmhbCwg2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCi  
gZmlsbD0iIzY0NzQ4YiI+MS4gT25seSbLbnZpcm9ubWvdCBkb21haW4gdXNlZDwvdGV4d4KICA8  
dGV4dCB4PSI1NTUiIHk9IjEzMCigZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250L  
XNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj4yLiBNaxNzaW5nIHN0b3JhZ2UgZG9tYwluPC90ZXh0Pg  
ogIDx0ZXh0IHg9IjU1NSIgeT0iMTQ1IIiBmb250LWzbhWlseT0iQXJpYwvsiHNhbnMtc2VyaWYiIGZ  
vbnQtc2l6ZT0iMTAiIGZpbGw9IiM2NDc00GIiPjMuIFNlZ21lbnQgbm90IHNlbGVjdGVkPC90ZXh0  
PgogIDx0ZXh0IHg9IjU1NSIgeT0iMTYwIiBmb250LWzbhWlseT0iQXJpYwvsiHNhbnMtc2VyaWYiI  
GZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM2NDc00GIiPjQuIFNlY3VyaXR5IGNvbnRleHQgbWlzc2luZz  
wvdGV4d4KICA8dGV4dCB4PSI1NTUiIHk9IjE3NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXN  
lcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIj41LiBhcmFpbCBkYXRhIG5vdCBpbmR1  
eGVkPC90ZXh0Pg0KICA8cmVjdCB4PSI1NTUiIHk9IjE4MiIg21kdGg9IjIwMCiGaGVpZ2h0PSIxM  
iIgcng9IjIiIGZpbGw9IiNLZGU5ZmuLz4KICA8dGV4dCB4PSI2NTUiIHk9IjE5MiIgZm9udC1mYW  
1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNmq0GQ5IiB0ZXh  
0LWFuY2hvcj0ibWlkZgxlij5GaXg6IEFkZCbdG9yYwdll3NldHRpbmdzIGRvbWFpbm8L3RleHQ+  
CgogIDwhLS0gRGVidWdnaW5nIFN0ZXbzIFNlY3Rpb24gLS0+CiAgPHJlY3QgeD0iMzAiiHk9IjIxN  
SIg21kdGg9IjIc0MCiGaGVpZ2h0PSIxNTAiIHJ4PSI2IiBmaWxsPSIjZmZmIiBzdHJva2U9IiNLmm  
U4ZjAiIHN0cm9rZS13aWR0aD0iMiIvPgogIDx0ZXh0IHg9IjQwMCiGeT0iMjM4IiBmb250LWzbhWl  
seT0iQXJpYwvsiHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBm  
aWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0ibWlkZgxlij5TeXN0ZW1hdGljIERlyvNvZ2luZyBTdGVwc  
zwvdGV4d4KCiAgPCEtLSBTdGVwIGJveGVzIC0tPgogIDxyZWN0IHg9IjUwIiB5PSIyNTUiIHdpZH  
RoPSIxMjAiIghlaWdodD0iNTAiIHJ4PSI2IiBmaWxsPSJ1cmwoI2NoZwNrR3JhZCkiIGZpbHrlcj0  
idXjsKCN0c1NoYwRvdykiLz4KICA8dGV4dCB4PSIxMTAiIHk9IjI3NSIgZm9udC1mYW1pbHk9IkFy

aWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpGUiiHrleHQtYW5jaG9yPSJtaWRkbGUipjEuIFZlcmlmeSBHcm91cDwvdGV4dD4KICA8dGV4dB4PSIxMTAiIHk9IjI5MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAu0SkiiHrleHQtYW5jaG9yPSJtaWRkbGUipkzlzIHVzZXIgaW4gZ3JvdXA/PC90ZXh0PgoKICA8cGF0aCBkPSJNMTcwLDI4MCBMMTkwLDI4MCiGc3Ryb2tlPSIjNjQ3NDhiIiBzdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmuig1hcmtlc1lbbmQ9InVybCgjdHNbcnJvdykiLz4KCiAgPHJly3QgeD0iMjAwIiB5PSIyNTUiIHdpZHRoPSIxMjAiIGHlaWdodD0iNTAiIHJ4PSI2IiBmaWxsPSJ1cmwoI2NoZWNrR3JhZCkiIGZpbHRlcj0idXjsKCN0c1NoYWrvdykiLz4KICA8dGV4dCB4PSIyNjAiIHk9IjI3NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpGUiiHrleHQtYW5jaG9yPSJtaWRkbGUipjIuIENoZWNrIFBvbGljeTwvdGV4dD4KICA8dGV4dCB4PSIyNjAiIHk9IjI5MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAu0SkiiHrleHQtYW5jaG9yPSJtaWRkbGUipLbvbGljeSBib3VuZD88L3RleHQ+CgogIDxwYXR0IGQ9Ik0zMjAsMjgwIEwzNDAsMjgwIiBzdHJva2U9IiM2NDc00GIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXjsKCN0c0Fycm93KSiPgoKICA8cmVjdCB4PSIzNTAiIHk9IjI1NSIgd2lkdGg9IjEyMCiGaGVpz2h0PSI1MCiGcng9IjYiIGZpbGw9InVybCgjY2h1Y2tHcmFkKSigZmlsdGVyPSJ1cmwoI3RzU2hhZG93KSiPgoIDx0ZXh0IHg9IjQxMCiGeT0iMjC1iBmb250LWZhbwlseT0iQXjpYwlsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aG10ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+My4gUmV2aWV3IEJvdW5kYXJ5PC90ZXh0PgoIDx0ZXh0IHg9IjQxMCiGeT0iMjkyIiBmb250LWZhbwlseT0iQXjpYwlsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+QWxsIDMgZG9tYwLucz88L3RleHQ+CgogIDxwYXR0IGQ9Ik00NzAsMjgwIEw00TAsMjgwIiBzdHJva2U9IiM2NDc00GIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyLWVuZD0idXjsKCN0c0Fycm93KSiPgoKICA8cmVjdCB4PSI1MDAiIHk9IjI1NSIgd2lkdGg9IjEyMCiGaGVpz2h0PSI1MCiGcng9IjYiIGZpbGw9InVybCgjY2h1Y2tHcmFkKSigZmlsdGVyPSJ1cmwoI3RzU2hhZG93KSiPgoIDx0ZXh0IHg9IjU2MCiGeT0iMjC1iBmb250LWZhbwlseT0iQXjpYwlsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aG10ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+NC4gVGVzdCBhcyBBZG1pbjwvdGV4dD4KICA8dGV4dCB4PSI1NjAiIHk9IjI5MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAu0SkiiHrleHQtYW5jaG9yPSJtaWRkbGUipkRhGEgZXhpc3RzPzwvdGV4dD4KCiAgPHBhdGggZD0iTTYyMCwy0DAgTDY0MCwy0DAiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lIiBtYXJrZXItZw5kPSJ1cmwoI3RzQXJyb3cpIi8+CgogIDxyZW0IHg9IjY1MCiGeT0iMjU1IiB3aWR0aD0iMTEwIiBoZWlnaHQ9IjUwIiByeD0iNiIgZmlsbD0idXjsKCNmaXhHcmFkKSigZmlsdGVyPSJ1cmwoI3RzU2hhZG93KSiPgoIDx0ZXh0IHg9IjcwNSIgeT0iMjC1iBmb250LWZhbwlseT0iQXjpYwlsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aG10ZSIgdGV4dC1hbmNb3I9Im1pZGRsZSI+NS4gQXBwbHkgRml4PC90ZXh0PgoIDxyZw0IHg9IjUwIiB5PSIzMjAiIhdPZHRoPSI3MDAiIGHlaWdodD0iMzAiiHJ4PSI0IiBmaWxsPSIjZTBmMmZlIi8+CiAgPHRleHqgeD0iNDAwIiB5PSIzNDAiIGZvbNqtZmFtaWx5PSJBcmhbCwgcfucy1zZXJpZiIgZm9udC1zaXplPSIxMCiGZmlsbD0iIzAzNj1hMSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+UHJvIFRpcoDogQwx3YXlzIHRlc3Qgd2l0aCBhIHRlc3QgdXNlciBhY2NvdW50LCBub3QgYWRtaW4uIEFkbWluIG1heSBoYXZlIGJyb2FkZXIgYWNjZXNzIHRoYXQgbWFza3MgaXNzdWVzLjwvdGV4dD4KPC9zdmc+Cg==)

### ### Issue 1: User Can't See Any Data

#### \*\*Symptoms\*\*:

- Apps show empty results
- DQL queries return no records

#### \*\*Troubleshooting\*\*:

| Check             | Action                                    |
|-------------------|-------------------------------------------|
| Group membership  | Verify user is in correct group           |
| Policy binding    | Verify group has policy bound             |
| Boundary scope    | Verify boundary isn't too restrictive     |
| Data policy       | Verify Data Viewer/Editor policy included |
| Segment selection | Check if restrictive segment is active    |

#### \*\*Resolution Steps\*\*:

1. Check user's group membership in Account Management
2. Verify group has appropriate policy binding
3. Review boundary conditions
4. Test with admin account to verify data exists

### ### Issue 2: User Sees Too Much Data

#### \*\*Symptoms\*\*:

- User sees data outside their scope
- Cross-team data visible

#### \*\*Troubleshooting\*\*:

| Check                     | Action                                   |
|---------------------------|------------------------------------------|
| Boundary conditions       | Verify boundary is correctly restrictive |
| Multiple group membership | User may be in multiple groups           |
| Security context          | Verify entities have correct context     |
| Policy conditions         | Check for overly broad conditions        |

#### \*\*Resolution Steps\*\*:

1. Review all groups user belongs to
2. Check all policy bindings for those groups
3. Verify boundary conditions are correct
4. Check security context on visible entities

### ### Issue 3: Segment Filter Not Working

#### \*\*Symptoms\*\*:

- Segment selected but data not filtered
- Same data visible with/without segment

**\*\*Troubleshooting\*\*:**

| Check           | Action                             |
|-----------------|------------------------------------|
| Segment syntax  | Verify DQL filter is correct       |
| Tag existence   | Verify entities have expected tags |
| App support     | Verify app supports segments       |
| Segment sharing | Verify segment is shared with user |

**\*\*Resolution Steps\*\*:**

1. Test segment filter as standalone DQL query
2. Verify tags exist on target entities
3. Check segment is shared/public
4. Try segment in different app

**### Issue 4: Classic Apps Work, New Apps Don't**

**\*\*Symptoms\*\*:**

- Legacy UI shows expected data
- New apps (Services, Logs) show no/wrong data

**\*\*Cause\*\*:** MZ-based access works for classic, but new apps need segments

**\*\*Resolution\*\*:**

1. Create corresponding segment for MZ filter
2. Ensure user selects segment in new apps
3. Consider setting default segment for user/group

---

**## 6. Diagnostic Queries**

**### Check Entity Tag Distribution**

```
```dql
// Analyze tag patterns for segment creation
fetch dt.entity.service
| expand tag = tags
| parse tag, "LD:key ':' LD:value"
| summarize count = count(), by:{key}
| sort count desc
| limit 20
```

```

```
```dql
// Find entities without standard tags
// These may need tagging for segment filters
```

```

```

fetch dt.entity.service
| filter NOT matchesValue(tags, "team:*")
| filter NOT matchesValue(tags, "env:*")
| fields entity.name, tags
| limit 50
```

### Check MZ Assignment Health

```dql
// Entities with MZ but no security context
// These need security context for boundary-based access
fetch dt.entity.service
| filter arraySize(managementZones) > 0
| filter isNull(dt.security_context)
| fields entity.name, managementZones, tags
| limit 50
```

```dql
// MZ to tag alignment check
// Verify MZ membership aligns with tags used in segments
fetch dt.entity.service
| expand mz = managementZones
| summarize
 entityCount = count(),
 withTeamTag = countIf(matchesValue(tags, "team:*")),
 withEnvTag = countIf(matchesValue(tags, "env:*")),
 by:{mz}
| sort entityCount desc
```

---

## 7. Ongoing Maintenance

### Regular Validation Tasks



Task	Frequency	Purpose
Security context audit	Weekly	Ensure new entities have context
Segment effectiveness	Monthly	Verify segments return expected data
Access review	Quarterly	Verify user access is appropriate
Policy review	Quarterly	Ensure policies match requirements



### New Entity Onboarding

When new services/hosts are deployed:

```

1. ****Auto-tagging**:** Ensure auto-tagging rules apply correct tags
2. ****Security context**:** Verify security context is set
3. ****Segment inclusion**:** Verify entity appears in appropriate segments
4. ****Access verification**:** Test user access to new entity

New User Onboarding

When new users join:

1. Add user to appropriate group(s)
2. Communicate available segments
3. Provide access documentation
4. Verify access works as expected

8. Health Monitoring Queries

Create Monitoring Dashboard

Use these queries for an access health dashboard:

```
```dql
// KPI: Security Context Coverage
fetch dt.entity.service
| summarize
 total = count(),
 covered = countIf(isNotNull(dt.security_context))
| fields coveragePercent = round(100.0 * covered / total, decimals: 1)
```

```

```
```dql
// KPI: Entities per Security Context
// Monitor for unexpected growth/decline
fetch dt.entity.service
| summarize count = count(), by:{dt.security_context}
| sort count desc
| limit 10
```

```

```
```dql
// KPI: Untagged Entities (migration gaps)
fetch dt.entity.service
| filter isNull(dt.security_context)
| filter NOT matchesValue(tags, "team:*)
| summarize untaggedCount = count()
```

```

```
----  
  
## 9. Best Practices Summary  
  
### Access Control  
  
- ✓ Use groups, not individual user permissions  
- ✓ Apply least privilege principle  
- ✓ Document all policy bindings  
- ✓ Regular access reviews  
  
### Segments  
  
- ✓ Align with organizational structure  
- ✓ Use consistent naming conventions  
- ✓ Test filters before deployment  
- ✓ Share segments appropriately  
  
### Boundaries  
  
- ✓ Create reusable boundaries  
- ✓ Use security context for entity-level control  
- ✓ Document boundary purpose  
- ✓ Keep conditions simple  
  
### Maintenance  
  
- ✓ Automate security context assignment  
- ✓ Monitor access health metrics  
- ✓ Review and update documentation  
- ✓ Train new team members  
  
----  
  
## Summary  
  
In this notebook, you learned:  
  
1. **Validation procedures**: How to verify migration success  
2. **Troubleshooting**: Common issues and resolutions  
3. **Diagnostic queries**: DQL for investigating access problems  
4. **Ongoing maintenance**: Regular tasks for access health  
5. **Health monitoring**: Queries for access dashboards  
  
## Migration Complete!  
  
Congratulations on completing the MZ2POL migration series. You have:
```

- Understood why migration is necessary (MZ2POL-01)
- Learned the new access control model (MZ2POL-02)
- Assessed and planned your migration (MZ2POL-03)
- Configured policies and boundaries (MZ2POL-04)
- Implemented segments (MZ2POL-05)
- Executed the migration (MZ2POL-06)
- Validated and can troubleshoot (MZ2POL-07)

Additional Resources

- [Access Management Concepts]
(<https://docs.dynatrace.com/docs/manage/identity-access-management/permission-management/access-concepts>)
- [Dynatrace Community – Boundaries + Policies Best Practices]
(<https://community.dynatrace.com/t5/Open-Q-A/Boundaries-Policies-Best-Practices-Advice/td-p/284808>)
- [Segments Documentation]
(<https://docs.dynatrace.com/docs/manage/segments/concepts/segments-concepts-queries>)
- [IAM Policy Reference] (<https://docs.dynatrace.com/docs/manage/identity-access-management/permission-management/manage-user-permissions-policies/advanced/iam-policystatements>)