```
# Organizing Your Environment

> **Series:** ONBRD | **Notebook:** 6 of 10 | **Created:** December 2025

## Tags, Segments, and Naming Conventions

As your Dynatrace environment grows, organization becomes critical. This
notebook covers how to structure your environment with tags, segments, and
naming conventions for maintainability and access control.

---

## Table of Contents

1. Why Organization Matters
2. Modern Organization Building Blocks
3. Tagging with Host Properties and Cloud Tags
4. Segments for Data Filtering
5. Naming Conventions
6. Querying by Tags and Properties
7. Next Steps

---

## Prerequisites

- Admin or Configurator access to Dynatrace
- Entities discovered (hosts, services)
- Understanding of your organizational structure

## 1. Why Organization Matters

Without organization, Dynatrace environments become difficult to manage:

| Problem | Impact |
|---------|--------|
| **No structure** | Can't find entities quickly |
| **No ownership** | Don't know who to contact |
| **No access control** | Everyone sees everything |
| **No filtering** | Dashboards show irrelevant data |
| **No grouping** | Can't compare similar systems |

### Modern Organization Building Blocks

![Organization Hierarchy]
(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA1MDAgMzQwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
```

```
icG9sR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAgICAgIDxzdG9w
IG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiM4YjVjZjY7c3RvcC1vcGFjaXR5OjEiIC8+C
iAgICAgIDxzdG9wIG9mZnNldD0iMTAwJSIgc3R5bGU9InN0b3AtY29sb3I6IzdjM2FlZDtzdG9wLW
9wYWNpdHk6MSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGllbnQgaWQ
9InNlZ0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgogICAgICA8c3Rv
cCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojM2I4MmY2O3N0b3Atb3BhY2l0eToxIiAvP
gogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMyNTYzZWI7c3RvcC
1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW50IGl
kPSJ0YWdHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAgPHN0
b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzEwYjk4MTtzdG9wLW9wYWNpdHk6MSIgL
z4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMDU5NjY5O3N0b3
Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCB
pZD0ibmFtZUdyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUiPgogICAgICA8
c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojZjU5ZTBiO3N0b3Atb3BhY2l0eToxI
iAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNkOTc3MDY7c3
RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmVhckdyYWRpZW5
0IGlkPSJlbnRHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzY0NzQ4YjtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojNDc1NTY5O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxmaWx0ZXIgaWQ9Im9
yZ1NoYWRvdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjEiIGR5PSIxIiBzdGREZXZpYXRpb249
IjIiIGZsb29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogIDwvZGVmcz4KCiAgPCEtL
SBCYWNrZ3JvdW5kIC0tPgogIDxyZWN0IHdpZHRoPSI1MDAiIGhlaWdodD0iMzQwIiBmaWxsPSIjZj
hmOWZhIiByeD0iMTAiLz4KCiAgPCEtLSBUaXRsZSAtLT4KICA8dGV4dCB4PSIyNTAiIHk9IjI4IiB
mb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTYiIGZvbnQtd2VpZ2h0
PSJib2xkIiBmaWxsPSIjMzMzIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Pcmdhbml6YXRpb24gSGllc
mFyY2h5IChNb2Rlcm4gUGxhdGZvcm0pPC90ZXh0PgoKICA8IS0tIExheWVyIDE6IFBvbGljaWVzIC
YgUGVybWlzc2lvbnMgLS0+CiAgPHJlY3QgeD0iNDAiIHk9IjUwIiB3aWR0aD0iNDIwIiBoZWlnaHQ
9IjQ4IiByeD0iOCIgZmlsbD0idXJsKCNwb2xHcmFkKSIgZmlsdGVyPSJ1cmwoI29yZ1NoYWRvdyki
Lz4KICA8dGV4dCB4PSIyNTAiIHk9IjcyIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiI
GZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3
I9Im1pZGRsZSI+UG9saWNpZXMgJmFtcDsgUGVybWlzc2lvbnM8L3RleHQ+CiAgPHRleHQgeD0iMjU
wIiB5PSI4OCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBm
aWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPihBY2Nlc3MgY
29udHJvbCB2aWEgSUFNKTwvdGV4dD4KCiAgPCEtLSBMYXllciAyOiBTZWdtZW50cyAtLT4KICA8cm
VjdCB4PSI0MCIgeT0iMTA1IiB3aWR0aD0iNDIwIiBoZWlnaHQ9IjQ4IiByeD0iOCIgZmlsbD0idXJ
sKCNzZWdHcmFkKSIgZmlsdGVyPSJ1cmwoI29yZ1NoYWRvdykiLz4KICA8dGV4dCB4PSIyNTAiIHk9
IjEyNyIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEyIiBmb250L
XdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlNlZ21lbnRzPC
90ZXh0PgogIDx0ZXh0IHg9IjI1MCIgeT0iMTQzIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2V
yaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNo
b3I9Im1pZGRsZSI+KERRTC1iYXNlZCBkYXRhIGZpbHRlcmluZyk8L3RleHQ+CgogIDwhLS0gTGF5Z
XIgMzogVGFncyAtLT4KICA8cmVjdCB4PSI0MCIgeT0iMTYwIiB3aWR0aD0iNDIwIiBoZWlnaHQ9Ij
Q4IiByeD0iOCIgZmlsbD0idXJsKCN0YWdHcmFkKSIgZmlsdGVyPSJ1cmwoI29yZ1NoYWRvdykiLz4
KICA8dGV4dCB4PSIyNTAiIHk9IjE4MiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBm
b250LXNpemU9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yP
SJtaWRkbGUiPlRhZ3MgKEhvc3QgUHJvcGVydGllcyArIENsb3VkIFRhZ3MpPC90ZXh0PgogIDx0ZX
h0IHg9IjI1MCIgeT0iMTk4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l
```

```
6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+
KEZsZXhpYmxlIGdyb3VwaW5nLCBzZXQgYXQgc291cmNlKTwvdGV4dD4KCiAgPCEtLSBMYXllciA0O
iBOYW1pbmcgQ29udmVudGlvbnMgLS0+CiAgPHJlY3QgeD0iNDAiIHk9IjIxNSIgd2lkdGg9IjQyMC
IgaGVpZ2h0PSI0OCIgcng9IjgiIGZpbGw9InVybCgjbmFtZUdyYWQpIiBmaWx0ZXI9InVybCgjb3J
nU2hhZG93KSIvPgogIDx0ZXh0IHg9IjI1MCIgeT0iMjM3IiBmb250LWZhbWlseT0iQXJpYWwsIHNh
bnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgd
GV4dC1hbmNob3I9Im1pZGRsZSI+TmFtaW5nIENvbnZlbnRpb25zPC90ZXh0PgogIDx0ZXh0IHg9Ij
I1MCIgeT0iMjUzIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTA
iIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+KENvbnNp
c3RlbmN5LCBkaXNjb3ZlcmFiaWxpdHkpPC90ZXh0PgoKICA8IS0tIExheWVyIDU6IEVudGl0aWVzI
C0tPgogIDxyZWN0IHg9IjQwIiB5PSIyNzAiIHdpZHRoPSI0MjAiIGhlaWdodD0iNDgiIHJ4PSI4Ii
BmaWxsPSJ1cmwoI2VudEdyYWQpIiBmaWx0ZXI9InVybCgjb3JnU2hhZG93KSIvPgogIDx0ZXh0IHg
9IjI1MCIgeT0iMjkyIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0i
MTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+R
W50aXRpZXM8L3RleHQ+CiAgPHRleHQgeD0iMjUwIiB5PSIzMDgiIGZvbnQtZmFtaWx5PSJBcmlhbC
wgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB
0ZXh0LWFuY2hvcj0ibWlkZGxlIj4oSG9zdHMsIHNlcnZpY2VzLCBwcm9jZXNzZXMpPC90ZXh0PgoK
ICA8IS0tIEFycm93IGluZGljYXRvciBvbiBsZWZ0IC0tPgogIDx0ZXh0IHg9IjIyIiB5PSIxODIiI
GZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxOCIgZmlsbD0iIzY0Nz
Q4YiI+4payPC90ZXh0PgogIDx0ZXh0IHg9IjE4IiB5PSIyMDAiIGZvbnQtZmFtaWx5PSJBcmlhbCw
gc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4YiIgd3JpdGluZy1tb2RlPSJ0
YiI+QWNjZXNzPC90ZXh0Pgo8L3N2Zz4K)
```

```
> **Note:** The modern Dynatrace platform uses **Segments** for data
filtering and **Policies** for access control. This replaces the legacy
Management Zones approach.

## 2. Modern Organization Building Blocks

The modern Dynatrace platform (Gen3/Grail) uses a "tag at source" approach
rather than rule-based auto-tagging:

### Tag Sources

| Source | How It Works | Best For |
|--------|--------------|----------|
| **Host Properties** | Set via OneAgent configuration | Environment, team,
tier metadata |
| **Cloud Provider Tags** | Automatically imported from AWS/Azure/GCP | Cloud
resource organization |
| **Kubernetes Labels** | Imported from K8s metadata | Container workload
organization |
| **OpenTelemetry Attributes** | Set in instrumentation | Custom service
attributes |

### The "Enrich at Source" Philosophy
```

```
![Modern Tagging Flow]
(data:image/svg+xml;base64,PHN2ZyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdm
ciIHZpZXdCb3g9IjAgMCA3MDAgMjIwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iaW5mcmFHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAgPHN0
b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzY0NzQ4YjtzdG9wLW9wYWNpdHk6MSIgL
z4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojNDc1NTY5O3N0b3
Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWVudCB
pZD0iYWdlbnRHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAg
PHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzEwYjk4MTtzdG9wLW9wYWNpdHk6M
SIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMDU5NjY5O3
N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxsaW5lYXJHcmFkaWV
udCBpZD0iZ3JhaWxHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAg
ICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzE0OTZmZjtzdG9wLW9wYWNpd
Hk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMGE2NG
JjO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICAgIDxmaWx0ZXIgaWQ
9InRhZ1NoYWRvdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjEiIGR5PSIxIiBzdGREZXZpYXRp
b249IjIiIGZsb29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsdGVyPgogICAgPG1hcmtlciBpZ
D0idGFnQXJyb3ciIG1hcmtlcldpZHRoPSIxMCIgbWFya2VySGVpZ2h0PSI3IiByZWZYPSI5IiByZW
ZZPSIzLjUiIG9yaWVudD0iYXV0byI+CiAgICAgIDxwb2x5Z29uIHBvaW50cz0iMCAwLCAxMCAzLjU
sIDAgNyIgZmlsbD0iIzY0NzQ4YiIvPgogICAgPC9tYXJrZXI+CiAgPC9kZWZzPgoKICA8IS0tIEJh
Y2tncm91bmQgLS0+CiAgPHJlY3Qgd2lkdGg9IjcwMCIgaGVpZ2h0PSIyMjAiIGZpbGw9IiNmOGY5Z
mEiIHJ4PSIxMCIvPgoKICA8IS0tIFRpdGxlIC0tPgogIDx0ZXh0IHg9IjM1MCIgeT0iMjgiIGZvbn
QtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxNiIgZm9udC13ZWlnaHQ9ImJ
vbGQiIGZpbGw9IiMzMzMiIHRleHQtYW5jaG9yPSJtaWRkbGUiPk1vZGVybiBUYWdnaW5nIEZsb3c8
L3RleHQ+CiAgPHRleHQgeD0iMzUwIiB5PSI0NiIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlc
mlmIiBmb250LXNpemU9IjExIiBmaWxsPSIjNjY2IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4iRW5yaW
NoIGF0IFNvdXJjZSIgUGhpbG9zb3BoeTwvdGV4dD4KCiAgPCEtLSBJbmZyYXN0cnVjdHVyZSBCb3g
gLS0+CiAgPHJlY3QgeD0iNDAiIHk9IjcwIiB3aWR0aD0iMTYwIiBoZWlnaHQ9IjgwIiByeD0iMTAi
IGZpbGw9InVybCgjaW5mcmFHcmFkKSIgZmlsdGVyPSJ1cmwoI3RhZ1NoYWRvdykiLz4KICA8dGV4d
CB4PSIxMjAiIHk9IjEwMCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpem
U9IjEyIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGU
iPkluZnJhc3RydWN0dXJlPC90ZXh0PgogIDx0ZXh0IHg9IjEyMCIgeT0iMTE4IiBmb250LWZhbWls
eT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyN
TUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+KENsb3VkL0s4cyk8L3RleHQ+CgogIDwhLS0gQX
Jyb3cgMSAtLT4KICA8cGF0aCBkPSJNMjA1LDExMCBMMjU1LDExMCIgc3Ryb2tlPSIjNjQ3NDhiIiB
zdHJva2Utd2lkdGg9IjIiIGZpbGw9Im5vbmUiIG1hcmtlci1lbmQ9InVybCgjdGFnQXJyb3cpIi8+
CgogIDwhLS0gT25lQWdlbnQgQm94IC0tPgogIDxyZWN0IHg9IjI2MCIgeT0iNzAiIHdpZHRoPSIxN
jAiIGhlaWdodD0iODAiIHJ4PSIxMCIgZmlsbD0idXJsKCNhZ2VudEdyYWQpIiBmaWx0ZXI9InVybC
gjdGFnU2hhZG93KSIvPgogIDx0ZXh0IHg9IjM0MCIgeT0iMTAwIiBmb250LWZhbWlseT0iQXJpYWw
sIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0
ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+T25lQWdlbnQ8L3RleHQ+CiAgPHRleHQgeD0iMzQwIiB5P
SIxMTgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD
0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj4rIEhvc3QgUHJvcGV
ydGllczwvdGV4dD4KCiAgPCEtLSBBcnJvdyAyIC0tPgogIDxwYXRoIGQ9Ik00MjUsMTEwIEw0NzUs
MTEwIiBzdHJva2U9IiM2NDc0OGIiIHN0cm9rZS13aWR0aD0iMiIgZmlsbD0ibm9uZSIgbWFya2VyL
WVuZD0idXJsKCN0YWdBcnJvdykiLz4KCiAgPCEtLSBEeW5hdHJhY2UvR3JhaWwgQm94IC0tPgogID
xyZWN0IHg9IjQ4MCIgeT0iNzAiIHdpZHRoPSIxNjAiIGhlaWdodD0iODAiIHJ4PSIxMCIgZmlsbD0
idXJsKCNncmFpbEdyYWQpIiBmaWx0ZXI9InVybCgjdGFnU2hhZG93KSIvPgogIDx0ZXh0IHg9IjU2
MCIgeT0iMTAwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiI
GZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RHluYX
RyYWNlPC90ZXh0PgogIDx0ZXh0IHg9IjU2MCIgeT0iMTE4IiBmb250LWZhbWlseT0iQXJpYWwsIHN
hbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4
dC1hbmNob3I9Im1pZGRsZSI+KEdyYWlsKTwvdGV4dD4KCiAgPCEtLSBMYWJlbHMgYmVsb3cgYm94Z
XMgLS0+CiAgPHRleHQgeD0iMTIwIiB5PSIxNzAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZX
JpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzY0NzQ4YiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+Q2x
vdWQgdGFnczwvdGV4dD4KICA8dGV4dCB4PSIxMjAiIHk9IjE4MyIgZm9udC1mYW1pbHk9IkFyaWFs
LCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIiB0ZXh0LWFuY2hvcj0ib
WlkZGxlIj5LOHMgbGFiZWxzPC90ZXh0PgoKICA8dGV4dCB4PSIzNDAiIHk9IjE3MCIgZm9udC1mYW
1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIiB0ZXh
0LWFuY2hvcj0ibWlkZGxlIj5Ib3N0IHByb3BlcnRpZXM8L3RleHQ+CiAgPHRleHQgeD0iMzQwIiB5
PSIxODMiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsb
D0iIzY0NzQ4YiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+cGFzc2VkIHRocm91Z2g8L3RleHQ+CgogID
x0ZXh0IHg9IjU2MCIgeT0iMTcwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQ
tc2l6ZT0iMTAiIGZpbGw9IiM2NDc0OGIiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlF1ZXJ5YWJsZTwv
dGV4dD4KICA8dGV4dCB4PSI1NjAiIHk9IjE4MyIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlc
mlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5hdH
RyaWJ1dGVzPC90ZXh0PgoKICA8IS0tIEJlbmVmaXQgY2FsbG91dCAtLT4KICA8cmVjdCB4PSIxNTA
iIHk9IjE5NSIgd2lkdGg9IjQwMCIgaGVpZ2h0PSIyMCIgcng9IjQiIGZpbGw9IiNkMWZhZTUiIHN0
cm9rZT0iIzEwYjk4MSIgc3Ryb2tlLXdpZHRoPSIxIi8+CiAgPHRleHQgeD0iMzUwIiB5PSIyMDkiI
GZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0iIzA0Nz
g1NyIgdGV4dC1hbmNob3I9Im1pZGRsZSI+U2FtZSB0YWdzIGF2YWlsYWJsZSBvbiBtZXRyaWNzLCB
sb2dzLCBzcGFucywgYW5kIGVudGl0aWVzPC90ZXh0Pgo8L3N2Zz4K)
```

~~~

Why "Tag at Source"?

Benefit	Description
Consistent	Same tags on metrics, logs, spans, and entities
Scalable	No processing overhead to apply rules
Traceable	Tags come from the source of truth
Real-time	No delay waiting for rule evaluation

3. Tagging with Host Properties and Cloud Tags

Host Properties (OneAgent)

Set custom properties during OneAgent installation or via configuration:

During Installation:
```bash
# Linux
sudo /bin/sh Dynatrace-OneAgent.sh \
  --set-host-property=env=production \
  --set-host-property=team=platform \
  --set-host-property=tier=backend
~~~

~~~
# Windows
.\Dynatrace-OneAgent.exe --set-host-property=env=production --set-host-property=team=checkout
```

Via Configuration File:
```
# /var/lib/dynatrace/oneagent/agent/config/hostcustomproperties.conf
env=production
team=platform
cost-center=engineering
```

Recommended Property Categories

Category	Example Properties	Purpose
Environment	`env=prod`, `env=staging`, `env=dev`	Distinguish environments
Owner	`team=platform`, `team=checkout`	Identify responsible team
Application	`app=ecommerce`, `app=mobile-api`	Group by application
Cost Center	`cost-center=marketing`	Financial allocation
Tier	`tier=frontend`, `tier=backend`	Architecture layer

Cloud Provider Tags

Cloud tags are automatically imported when using cloud integrations:

Cloud	Tag Format	DQL Field
AWS	AWS resource tags	`aws.tag.*`
Azure	Azure resource tags	`azure.tag.*`
GCP	GCP labels	`gcp.label.*`

AWS Tag Example:
If your EC2 instance has tag `Environment=Production`, it appears as `aws.tag.Environment` in Dynatrace.

Kubernetes Labels

K8s labels are automatically available for container workloads:

K8s Metadata	DQL Field
Namespace	`k8s.namespace.name`
Deployment	`k8s.deployment.name`
~~~

~~~
| Pod labels | `k8s.pod.labels.*` |
| Node labels | `k8s.node.labels.*` |

## 4. Segments for Data Filtering

Segments provide DQL-based filtering to create focused views of your data.

**Location:** Observe and explore → Segments

### What are Segments?

Segments are reusable DQL filters that:
- Filter data in Notebooks, Dashboards, and Apps
- Can be applied as default context
- Are shareable across the organization

### Creating a Segment

1. Go to Observe and explore → Segments
2. Click "Create segment"
3. Define your DQL filter:
   ```dql
   dt.entity.host.properties.env == "production"
   ```
4. Name your segment (e.g., "Production Environment")
5. Save

### Segment Use Cases

| Use Case | Segment Filter | Purpose |
|----------|---------------|---------|
| **Environment** | `properties.env == "prod"` | Focus on production |
| **Team** | `properties.team == "checkout"` | Team-specific view |
| **Application** | `service.name contains "payment"` | Application focus |
| **Region** | `aws.tag.Region == "us-east-1"` | Geographic filtering |

### Segments vs Legacy Management Zones

| Feature | Segments | Management Zones (Legacy) |
|---------|----------|---------------------------|
| **Filter basis** | DQL expressions | Rule-based matching |
| **Data types** | All Grail data | Entities only |
| **Flexibility** | Highly flexible | Limited rule types |
| **Access control** | Use Policies instead | Built-in |
| **Modern platform** | ✅ Recommended | ⚠️ Being deprecated |

### Segment Best Practices
~~~

Practice	Why
Use host properties	Consistent filtering
Name clearly	`Prod-Checkout-Team` not `Segment1`
Document purpose	Add description
Test filters	Verify expected data

## 5. Naming Conventions

Consistent naming makes entities discoverable and filtering effective.

### Host Naming

Set meaningful host names that encode key information:

Pattern	Example	Components
`{env}-{tier}-{seq}`	`prod-web-01`	Environment, tier, sequence
`{region}-{app}-{role}`	`us-east-ecom-api`	Region, app, role
`{team}-{service}-{id}`	`checkout-cart-a1b2`	Team, service, unique ID

### Host Naming via OneAgent

You can set a custom display name during installation:

```bash
sudo /bin/sh Dynatrace-OneAgent.sh --set-host-name="prod-web-01"
```

### Naming Principles

Principle	Good	Bad
Descriptive	`payment-service`	`svc-001`
Consistent	`prod-web-01`, `prod-web-02`	`prod-web-01`, `Web Server 2`
Parseable	`us-east-prod-checkout`	`USEastProdCheckout`
Unique	Include environment/region	Generic names

### Property Naming Standards

For host properties, use consistent naming:

Standard	Example	Why
Lowercase	`env=prod` not `ENV=PROD`	Consistency
Hyphen separated	`cost-center=eng`	Readability
Short keys	`env` not `environment`	Query simplicity

~~~markdown
| **Consistent values** | Always `prod` not sometimes `production` |
Filtering works |

## 6. Querying by Tags and Properties

Use host properties and cloud tags in DQL queries to filter and group data.

```dql
// Find hosts by name pattern
fetch dt.entity.host
| filter contains(entity.name, "prod")
| fields entity.name
| limit 20
```

```dql
// Count hosts by operating system type
fetch dt.entity.host
| summarize host_count = count(), by: {osType}
| sort host_count desc
```

```dql
// Find services by name pattern
fetch dt.entity.service
| filter contains(entity.name, "checkout")
| fields entity.name, serviceType
| limit 20
```

```dql
// Query logs filtered by host group (Kubernetes)
fetch logs, from: now() - 1h
| filter isNotNull(k8s.namespace.name)
| summarize log_count = count(), by: {k8s.namespace.name}
| sort log_count desc
| limit 20
```

```dql
// Query spans by service name pattern
fetch spans, from: now() - 1h
| filter span.kind == "server"
| filter contains(service.name, "payment")
| summarize request_count = count(), by: {service.name}
| sort request_count desc
| limit 20
```
~~~

~~~
```dql
// Find hosts by name pattern for environment identification
fetch dt.entity.host
| filter not(contains(entity.name, "prod"))
      and not(contains(entity.name, "staging"))
      and not(contains(entity.name, "dev"))
| fields entity.name
| limit 20
```

## 7. Next Steps

With organization in place:

1. **ONBRD-07: Understanding Your Data** - Explore what Dynatrace discovered
2. Define host properties for your environment
3. Create segments for team-specific views
4. Document your naming conventions

### Organization Checklist

- [ ] Host property strategy documented
- [ ] Properties set on OneAgent installations
- [ ] Cloud tags verified (if using cloud providers)
- [ ] Segments created for common filters
- [ ] Naming conventions established
- [ ] Access control configured via Policies (see ONBRD-02)

---

## Summary

In this notebook, you learned:

- Why organization matters for scalability
- The modern "tag at source" approach
- How to use host properties and cloud tags
- How to create and use Segments for filtering
- Naming convention best practices
- How to query by properties and attributes

---

## References

- [Host Properties](https://docs.dynatrace.com/docs/setup-and-
configuration/dynatrace-oneagent/installation-and-
~~~

```
operation/linux/installation/customize-oneagent-installation-on-linux)
- [Segments](https://docs.dynatrace.com/docs/observe-and-explore/segments)
- [Cloud Tags](https://docs.dynatrace.com/docs/setup-and-configuration/setup-
on-cloud-platforms)
- [Kubernetes Labels](https://docs.dynatrace.com/docs/ingest-from/setup-on-
k8s)
- [DQL Reference](https://docs.dynatrace.com/docs/platform/grail/dynatrace-
query-language)
```