```
# 🏢 Private Synthetic Locations

> **Series:** SYNTH | **Notebook:** 4 of 6 | **Created:** December 2025

## Monitoring Internal Applications from Your Infrastructure

This notebook covers deploying and managing private synthetic locations
(ActiveGates) for monitoring internal applications, APIs, and services not
accessible from the public internet.

---

## Table of Contents

1. Why Private Locations?
2. Architecture
3. Synthetic-Enabled ActiveGate
4. Deployment Options
5. Configuration
6. Monitoring Private Location Health
7. Troubleshooting


## Prerequisites

- ✅ Access to a Dynatrace environment with Synthetic Monitoring
- ✅ Completed SYNTH-01 through SYNTH-03
- ✅ Infrastructure access to deploy ActiveGate (for setup)



## 1. Why Private Locations?

### Public vs Private Locations

| Aspect | Public Locations | Private Locations |
|--------|-----------------|-------------------|
| Hosting | Dynatrace cloud | Your infrastructure |
| Access | Public internet only | Internal networks |
| Maintenance | Managed by Dynatrace | Managed by you |
| Security | External perspective | Behind firewall |
| Latency | Varies by region | Local network |

### Use Cases for Private Locations

| Scenario | Description |
|----------|-------------|
```

Internal APIs	Services not exposed to internet
VPN-only apps	Applications requiring VPN access
Pre-production	Staging/dev environments
Security compliance	Data must not leave network
Low latency testing	Local performance baselines
Isolated networks	Air-gapped or restricted networks

2. Architecture

Private synthetic locations use ActiveGates deployed within your infrastructure to execute monitors against internal applications:

![Private Location Architecture]
(
ciIHZpZXdCb3g9IjAgMCA4MDAgMzQwIj4KICA8ZGVmcz4KICAgIDxsaW5lYXJHcmFkaWVudCBpZD0
iaW5mcmFHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHkyPSIxMDAlIj4KICAgICAgPHN0
b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6I2VmNDQ0NDtzdG9wLW9wYWNpdHk6MC4xI
iAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiNkYzI2MjY7c3
RvcC1vcGFjaXR5OjAuMSIgLz4KICAgIDwvbGluZWFyR3JhZGllbnQ+CiAgICA8bGluZWFyR3JhZGl
lbnQgaWQ9ImNsb3VkR3JhZCIgeDE9IjAlIiB5MT0iMCUiIHgyPSIxMDAlIiB5Mj0iMTAwJSI+CiAg
ICAgIDxzdG9wIG9mZnNldD0iMCUiIHN0eWxlPSJzdG9wLWNvbG9yOiMzYjgyZjY7c3RvcC1vcGFja
XR5OjAuMSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC1jb2xvcjojMj
U2M2ViO3N0b3Atb3BhY2l0eTowLjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmV
hckdyYWRpZW50IGlkPSJhZ0dyYWQiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAwJSIgeTI9IjEwMCUi
PgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojOGI1Y2Y2O3N0b3Atb
3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPSJzdG9wLWNvbG9yOi
M3YzNhZWQ7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW50PgogICAgPGxpbmV
hckdyYWRpZW50IGlkPSJkdENsdXN0ZXJHcmFkIiB4MT0iMCUiIHkxPSIwJSIgeDI9IjEwMCUiIHky
PSIxMDAlIj4KICAgICAgPHN0b3Agb2Zmc2V0PSIwJSIgc3R5bGU9InN0b3AtY29sb3I6IzNiODJmN
jtzdG9wLW9wYWNpdHk6MSIgLz4KICAgICAgPHN0b3Agb2Zmc2V0PSIxMDAlIiBzdHlsZT0ic3RvcC
1jb2xvcjojMjU2M2ViO3N0b3Atb3BhY2l0eToxIiAvPgogICAgPC9saW5lYXJHcmFkaWVudD4KICA
gIDxsaW5lYXJHcmFkaWVudCBpZD0iYXBwR3JhZFByaXYiIHgxPSIwJSIgeTE9IjAlIiB4Mj0iMTAw
JSIgeTI9IjEwMCUiPgogICAgICA8c3RvcCBvZmZzZXQ9IjAlIiBzdHlsZT0ic3RvcC1jb2xvcjojM
TBiOTgxO3N0b3Atb3BhY2l0eToxIiAvPgogICAgICA8c3RvcCBvZmZzZXQ9IjEwMCUiIHN0eWxlPS
JzdG9wLWNvbG9yOiMwNTk2Njk7c3RvcC1vcGFjaXR5OjEiIC8+CiAgICA8L2xpbmVhckdyYWRpZW5
0PgogICAgPGZpbHRlciBpZD0icHJpdlNoYWRvdyI+CiAgICAgIDxmZURyb3BTaGFkb3cgZHg9IjEi
IGR5PSIxIiBzdGREZXZpYXRpb249IjIiIGZsb29kLW9wYWNpdHk9IjAuMTUiLz4KICAgIDwvZmlsd
GVyPgogICAgPG1hcmtlciBpZD0icHJpdkFycm93IiBtYXJrZXJXaWR0aD0iMTAiIG1hcmtlckhlaW
dodD0iNyIgcmVmWD0iOSIgcmVmWT0iMy41IiBvcmllbnQ9ImF1dG8iPgogICAgICA8cG9seWdvbiB
wb2ludHM9IjAgMCwgMTAgMy41LCAwIDciIGZpbGw9IiM2NDc0OGIiLz4KICAgIDwvbWFya2VyPgog
ICAgPG1hcmtlciBpZD0iaHR0cHNBcnJvdyIgbWFya2VyV2lkdGg9IjEwIiBtYXJrZXJIZWlnaHQ9I
jciIHJlZlg9IjkiIHJlZlk9IjMuNSIgb3JpZW50PSJhdXRvIj4KICAgICAgPHBvbHlnb24gcG9pbn
RzPSIwIDAsIDEwIDMuNSwgMCA3IiBmaWxsPSIjMTBiOTgxIi8+CiAgICA8L21hcmtlcj4KICA8L2R
lZnM+CgogIDwhLS0gQmFja2dyb3VuZCAtLT4KICA8cmVjdCB3aWR0aD0iODAwIiBoZWlnaHQ9IjM0
MCIgZmlsbD0iI2Y4ZjlmYSIgcng9IjEwIi8+CgogIDwhLS0gVGl0bGUgLS0+CiAgPHRleHQgeD0iN
DAwIiB5PSIyOCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE4Ii
Bmb250LXdlaWdodD0iYm9sZCIgZmlsbD0iIzMzMyIgdGV4dC1hbmNob3I9Im1pZGRsZSI+UHJpdmF
0ZSBTeW50aGV0aWMgTG9jYXRpb24gQXJjaGl0ZWN0dXJlPC90ZXh0PgogIDx0ZXh0IHg9IjQwMCIg

eT0iNDgiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZmlsb
D0iIzY2NiIgdGV4dC1hbmNob3I9Im1pZGRsZSI+TW9uaXRvciBpbnRlcm5hbCBhcHBsaWNhdGlvbn
Mgd2l0aG91dCBleHBvc2luZyB0aGVtIHRvIHRoZSBpbnRlcm5ldDwvdGV4dD4KCiAgPCEtLSBZb3V
yIEluZnJhc3RydWN0dXJlIEJveCAtLT4KICA8cmVjdCB4PSIzMCIgeT0iNjUiIHdpZHRoPSIzODAi
IGhlaWdodD0iMjMwIiByeD0iMTAiIGZpbGw9InVybCgjaW5mcmFHcmFkKSIgc3Ryb2tlPSIjZWY0N
DQ0IiBzdHJva2Utd2lkdGg9IjIiLz4KICA8dGV4dCB4PSIyMjAiIHk9IjkwIiBmb250LWZhbWlseT
0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWx
sPSIjZGMyNjI2IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5ZT1VSIElORlJBU1RSVUNUVVJFPC90ZXh0
PgoKICA8IS0tIEFdGl2ZUdhdGUgLS0+CiAgPHJlY3QgeD0iNTAiIHk9IjEwNSIgd2lkdGg9IjE4M
CIgaGVpZ2h0PSI5MCIgcng9IjgiIGZpbGw9InVybCgjYWdHcmFkKSIgZmlsdGVyPSJ1cmwoI3ByaX
ZTaGFkb3cpIi8+CiAgPHRleHQgeD0iMTQwIiB5PSIxMzAiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2F
ucy1zZXJpZiIgZm9udC1zaXplPSIxMSIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9IndoaXRlIiB0
ZXh0LWFuY2hvcj0ibWlkZGxlIj5TeW50aGV0aWMtRW5hYmxlZDwvdGV4dD4KICA8dGV4dCB4PSIxN
DAiIHk9IjE0OCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjExIi
Bmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkFjdGl
2ZUdhdGU8L3RleHQ+CiAgPHRleHQgeD0iMTQwIiB5PSIxNzUiIGZvbnQtZmFtaWx5PSJBcmlhbCwg
c2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIiB0Z
Xh0LWFuY2hvcj0ibWlkZGxlIj5TeW50aGV0aWMgRW5naW5lPC90ZXh0PgogIDx0ZXh0IHg9IjE0MC
IgeT0iMTg4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZ
pbGw9InJnYmEoMjU1LDI1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+KyBFeGVjdXRp
b24gUnVudGltZTwvdGV4dD4KCiAgPCEtLSBJbnRlcm5hbCBBcHBzIC0tPgogIDxyZWN0IHg9IjUwI
iB5PSIyMTAiIHdpZHRoPSIzNDAiIGhlaWdodD0iNzAiIHJ4PSI4IiBmaWxsPSIjZmZmIiBzdHJva2
U9IiMxMGI5ODEiIHN0cm9rZS13aWR0aD0iMiIvPgogIDx0ZXh0IHg9IjIyMCIgeT0iMjM1IiBmb25
0LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTEiIGZvbnQtd2VpZ2h0PSJi
b2xkIiBmaWxsPSIjMDQ3ODU3IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5JbnRlcm5hbCBBcHBsaWNhd
GlvbnM8L3RleHQ+CgogIDxyZWN0IHg9IjcwIiB5PSIyNTAiIHdpZHRoPSI3MCIgaGVpZ2h0PSIyMi
Igcng9IjQiIGZpbGw9InVybCgjYXBwR3JhZFByaXYpIi8+CiAgPHRleHQgeD0iMTA1IiB5PSIyNjU
iIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0id2hp
dGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkFQSXM8L3RleHQ+CgogIDxyZWN0IHg9IjE1MCIgeT0iM
jUwIiB3aWR0aD0iNzAiIGhlaWdodD0iMjIiIHJ4PSI0IiBmaWxsPSJ1cmwoI2FwcEdyYWRQcml2KS
IvPgogIDx0ZXh0IHg9IjE4NSIgeT0iMjY1IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWY
iIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5EYXRhYmFzZ
XM8L3RleHQ+CgogIDxyZWN0IHg9IjIzMCIgeT0iMjUwIiB3aWR0aD0iNzAiIGhlaWdodD0iMjIiIH
J4PSI0IiBmaWxsPSJ1cmwoI2FwcEdyYWRQcml2KSIvPgogIDx0ZXh0IHg9IjI2NSIgeT0iMjY1IiB
mb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IndoaXRl
IiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5JbnRyYW5ldDwvdGV4dD4KCiAgPHJlY3QgeD0iMzEwIiB5P
SIyNTAiIHdpZHRoPSI3MCIgaGVpZ2h0PSIyMiIgcng9IjQiIGZpbGw9InVybCgjYXBwR3JhZFByaX
YpIi8+CiAgPHRleHQgeD0iMzQ1IiB5PSIyNjUiIGZvbnQtZmFtaWx5PSJBcmlhbCwgc2Fucy1zZXJ
pZiIgZm9udC1zaXplPSIxMCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkxlZ2Fj
eTwvdGV4dD4KCiAgPCEtLSBBcnJvdyBmcm9tIEFHIHRvIEFwcHMgLS0+CiAgPHBhdGggZD0iTTE0MCwx
OTUgTDE0MCwyMDUiIHN0cm9rZT0iIzY0NzQ4YiIgc3Ryb2tlLXdpZHRoPSIyIiBmaWxsPSJub25lI
iBtYXJrZXItZW5kPSJ1cmwoI2ZyaXZBcnJvdykiLz4KICA8dGV4dCB4PSIxODUiIHk9IjIwNSIgZm
9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDh
iIj5Nb25pdG9yczwvdGV4dD4KCiAgPCEtLSBIVFRQUyBDb25uZWN0aW9uIC0tPgogIDxwYXRoIGQ9
Ik00MTUsMTUwIEw0NzUsMTUwIiBzdHJva2U9IiMxMGI5ODEiIHN0cm9rZS13aWR0aD0iMyIgZmlsb
D0ibm9uZSIgbWFya2VyLWVuZD0idXJsKCNodHRwc0Fycm93KSIvPgogIDx0ZXh0IHg9IjQ0NSIgeeT
0iMTQwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQ
td2VpZ2h0PSJib2xkIiBmaWxsPSIjMTBiOTgxIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5IVFRQUzwv

dGV4dD4KICA8dGV4dCB4PSI0NDUiIHk9IjE2NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlc
mlmIiBmb250LXNpemU9IjEwIiBmaWxsPSIjNjQ3NDhiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5PdX
Rib3VuZDwvdGV4dD4KCiAgPCEtLSBEeW5hdHJhY2UgQ2xvdWQgQm94IC0tPgogIDxyZWN0IHg9IjQ
4NSIgeT0iNjUiIHdpZHRoPSIyODUiIGhlaWdodD0iMjMwIiByeD0iMTAiIGZpbGw9InVybCgjY2xv
dWRHcmFkKSIgc3Ryb2tlPSIjM2I4MmY2IiBzdHJva2Utd2lkdGg9IjIiLz4KICA8dGV4dCB4PSI2M
jciIHk9IjkwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTIiIG
ZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMjU2M2ViIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5EWU5
BVFJBQ0UgQ0xVU1RFUjwvdGV4dD4KCiAgPCEtLSBDbHVzdGVyIENvbXBvbmVudHMgLS0+CiAgPHJl
Y3QgeD0iNTA1IiB5PSIxMTAiIHdpZHRoPSIxMTAiIGhlaWdodD0iNTAiIHJ4PSI2IiBmaWxsPSJ1c
mwoI2R0Q2x1c3RlckdyYWQpIiBmaWx0ZXI9InVybCgjcHJpdlNoYWRvdykiLz4KICA8dGV4dCB4PS
I1NjAiIHk9IjEzMyIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjE
wIiBmb250LXdlaWdodD0iYm9sZCIgZmlsbD0id2hpdGUiIHRleHQtYW5jaG9yPSJtaWRkbGUiPlNj
aGVkdWxpbmc8L3RleHQ+CiAgPHRleHQgeD0iNTYwIiB5PSIxNTAiIGZvbnQtZmFtaWx5PSJBcmlhb
Cwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZmlsbD0icmdiYSgyNTUsMjU1LDI1NSwwLjkpIi
B0ZXh0LWFuY2hvcj0ibWlkZGxlIj5UZXN0IHRyaWdnZXJzPC90ZXh0PgoKICA8cmVjdCB4PSI2NDA
iIHk9IjExMCIgd2lkdGg9IjExMCIgaGVpZ2h0PSI1MCIgcng9IjYiIGZpbGw9InVybCgjZHRDbHVz
dGVyR3JhZCkiIGZpbHRlcj0idXJsKCNwcml2U2hhZG93KSIvPgogIDx0ZXh0IHg9IjY5NSIgeT0iM
TMzIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2
VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+UmVzdWx0czwvdGV
4dD4KICA8dGV4dCB4PSI2OTUiIHk9IjE1MCIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlm
IiBmb250LXNpemU9IjEwIiBmaWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yP
SJtaWRkbGUiPlN0b3JhZ2U8L3RleHQ+CgogIDxyZWN0IHg9IjUwNSIgeT0iMTc1IiB3aWR0aD0iMT
EwIiBoZWlnaHQ9IjUwIiByeD0iNiIgZmlsbD0idXJsKCNkdENsdXN0ZXJHcmFkKSIgZmlsdGVyPSJ
1cmwoI3ByaXZTaGFkb3cpIi8+CiAgPHRleHQgeD0iNTYwIiB5PSIxOTgiIGZvbnQtZmFtaWx5PSJB
cmlhbCwgc2Fucy1zZXJpZiIgZm9udC1zaXplPSIxMCIgZm9udC13ZWlnaHQ9ImJvbGQiIGZpbGw9I
ndoaXRlIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5BbGVydGluZzwvdGV4dD4KICA8dGV4dCB4PSI1Nj
AiIHk9IjIxNSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmIiBmb250LXNpemU9IjEwIiB
maWxsPSJyZ2JhKDI1NSwyNTUsMjU1LDAuOSkiIHRleHQtYW5jaG9yPSJtaWRkbGUiPkRhdmlzIEFJ
PC90ZXh0PgoKICA8cmVjdCB4PSI2NDAiIHk9IjE3NSIgd2lkdGg9IjExMCIgaGVpZ2h0PSI1MCIgc
ng9IjYiIGZpbGw9InVybCgjZHRDbHVzdGVyR3JhZCkiIGZpbHRlcj0idXJsKCNwcml2U2hhZG93KS
IvPgogIDx0ZXh0IHg9IjY5NSIgeT0iMTk4IiBmb250LWZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWY
iIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSJ3aGl0ZSIgdGV4dC1hbmNo
b3I9Im1pZGRsZSI+QW5hbHl0aWNzPC90ZXh0PgogIDx0ZXh0IHg9IjY5NSIgeT0iMjE1IiBmb250L
WZhbWlseT0iQXJpYWwsIHNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9InJnYmEoMjU1LD
I1NSwyNTUsMC45KSIgdGV4dC1hbmNob3I9Im1pZGRsZSI+RGFzaGJvYXJkczwvdGV4dD4KCiAgPCE
tLSBLZXkgQmVuZWZpdHMgLS0+CiAgPHJlY3QgeD0iNTA1IiB5PSIyNDAiIHdpZHRoPSIyNDUiIGhl
aWdodD0iNDUiIHJ4PSI2IiBmaWxsPSIjZDFmYWU1IiBzdHJva2U9IiMxMGI5ODEiIHN0cm9rZS13a
WR0aD0iMSIvPgogIDx0ZXh0IHg9IjYyNyIgeT0iMjYwIiBmb250LWZhbWlseT0iQXJpYWwsIHNhbn
Mtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZvbnQtd2VpZ2h0PSJib2xkIiBmaWxsPSIjMDQ3ODU3IiB
0ZXh0LWFuY2hvcj0ibWlkZGxlIj5ObyBpbmJvdW5kIGZpcmV3YWxsIHJ1bGVzIG5lZWRlZDwvdGV4
dD4KICA8dGV4dCB4PSI2MjciIHk9IjI3NSIgZm9udC1mYW1pbHk9IkFyaWFsLCBzYW5zLXNlcmlmI
iBmb250LXNpemU9IjEwIiBmaWxsPSIjMDY0ZTNiIiB0ZXh0LWFuY2hvcj0ibWlkZGxlIj5Pbmx5IE
hUVFBTIG91dGJvdW5kIHRvIER5bmF0cmFjZTwvdGV4dD4KCiAgPCEtLSBCb3R0b20gbm90ZSAtLT4
KICA8cmVjdCB4PSIzMCIgeT0iMzA1IiB3aWR0aD0iNzQwIiBoZWlnaHQ9IjI1IiByeD0iNCIgZmls
bD0iI2ZlZjNjNyIvPgogIDx0ZXh0IHg9IjQwMCIgeT0iMzIyIiBmb250LWZhbWlseT0iQXJpYWwsI
HNhbnMtc2VyaWYiIGZvbnQtc2l6ZT0iMTAiIGZpbGw9IiM5MjQwMGUiIHRleHQtYW5jaG9yPSJtaW
RkbGUiPkRlcGxveSBTeW50aGV0aWMtRW5hYmxlZCBBY3RpdmVHYXRlIGluIHlvdXIgRE1aIG9yIGl
udGVybmFsIG5ldHdvcmsgdG8gbW9uaXRvciBwcml2YXRlIHJlc291cmNlczwvdGV4dD4KPC9zdmc+

```
Cg==)


### Key Points

- **Outbound only**: ActiveGate initiates all connections
- **No inbound ports**: No firewall changes for external access
- **Local execution**: Tests run inside your network
- **Results upload**: Only metrics/results sent to Dynatrace

## 3. Synthetic-Enabled ActiveGate

### ActiveGate Capabilities

ActiveGate can serve multiple purposes:

| Capability | Description |
|------------|-------------|
| **Synthetic** | Private synthetic location |
| **Metrics** | Metric ingestion endpoint |
| **Logs** | Log ingestion endpoint |
| **API** | Cluster API proxy |
| **Extensions** | Extension execution |

### Synthetic Engine Requirements

| Resource | Minimum | Recommended |
|----------|---------|-------------|
| **CPU** | 2 cores | 4+ cores |
| **RAM** | 4 GB | 8+ GB |
| **Disk** | 20 GB | 50+ GB |
| **Network** | HTTPS outbound | Low latency to targets |

### Browser Monitor Requirements

For browser monitors, additional requirements:
- Chrome/Chromium browser
- Display server (X11 or headless)
- Additional RAM for browser instances

## 4. Deployment Options

### Option 1: Linux/Windows Installer

```bash
# Download from Dynatrace Hub
# Settings → Deployment status → ActiveGate
```
```

```
# Linux installation
sudo /bin/sh Dynatrace-ActiveGate-Linux-x86-*.sh \
  --enable-synthetic

# Verify synthetic capability
sudo systemctl status dynatracegateway
```

Option 2: Container Deployment

```yaml
# docker-compose.yml
version: '3'
services:
  activegate:
    image: dynatrace/activegate:latest
    environment:
      - DT_TENANT=abc12345
      - DT_CAPABILITIES=synthetic
      - DT_API_TOKEN=${DT_API_TOKEN}
    volumes:
      - activegate-data:/var/lib/dynatrace/gateway
    ports:
      - "9999:9999"
```

Option 3: Kubernetes

```yaml
# Use Dynatrace Operator with ActiveGate CRD
apiVersion: dynatrace.com/v1beta1
kind: DynaKube
metadata:
  name: dynakube
spec:
  activeGate:
    capabilities:
      - synthetic-monitoring
    resources:
      requests:
        cpu: "500m"
        memory: "512Mi"
```

```dql
// List all synthetic locations
// Note: 'type' (PRIVATE/PUBLIC), 'status', 'city', 'countryCode' fields are
not available
```

```
// Private locations are typically identified by their naming convention
fetch dt.entity.synthetic_location
| fields id, entity.name
| sort entity.name asc
| limit 50
```

```dql
// Count synthetic locations
// Note: 'type' field not available on synthetic_location entity
// Use entity.name patterns to identify private vs public locations
fetch dt.entity.synthetic_location
| summarize {count = count()}
| fields count
```

5. Configuration

Creating a Private Location

1. **Deploy ActiveGate** with synthetic capability
2. **Navigate to**: Settings → Synthetic → Private synthetic locations
3. **Create location**: Name, description, geographic info
4. **Assign ActiveGates**: Select which ActiveGates serve this location

Location Settings

Setting	Description
Name	Descriptive location name
Latitude/Longitude	Geographic coordinates
City/Region	Location metadata
ActiveGate nodes	Assigned ActiveGates

High Availability

For production workloads:
- Deploy 2+ ActiveGates per location
- Distribute across availability zones
- Load balancing automatic

```dql
// Monitor executions by location
// Note: Private locations typically have custom names (not city names like
"N. Virginia")
// Identify your private locations by their naming convention
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
```

```
| summarize {
    executions = count(),
    availability_pct = round(countIf(synthetic.availability == true) * 100.0
/ count(), decimals: 2)
  }, by: {dt.entity.synthetic_test, dt.entity.synthetic_location}
| sort executions desc
| limit 30
```

6. Monitoring Private Location Health

ActiveGate Health Indicators

Metric	Description	Alert Threshold
Connectivity	Connection to cluster	Any disconnect
CPU Usage	Processing load	> 80% sustained
Memory	RAM utilization	> 80%
Disk	Storage usage	> 80%
Queue Depth	Pending executions	> 100

```dql
// Location availability over time
// Filter by your private location names using matchesPhrase if needed
fetch bizevents, from: now() - 7d
| filter event.provider == "dynatrace.synthetic"
| fieldsAdd hour_bucket = bin(timestamp, 1h)
| summarize {
    success_count = countIf(synthetic.availability == true),
    total_count = count()
  }, by: {dt.entity.synthetic_location, hour_bucket}
| fieldsAdd availability_pct = round((success_count * 100.0) / total_count,
decimals: 2)
| sort hour_bucket desc
```

```dql
// Execution count by location
// To filter for private locations, use entity.name patterns specific to your
naming convention
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| summarize {
    total_executions = count(),
    failed_executions = countIf(synthetic.availability == false),
    unique_monitors = countDistinct(dt.entity.synthetic_test)
  }, by: {dt.entity.synthetic_location}
| fieldsAdd failure_rate = round((failed_executions * 100.0) /
```

````
total_executions, decimals: 2)
| sort total_executions desc
```

## 7. Troubleshooting

### Common Issues

| Issue | Symptoms | Resolution |
|-------|----------|------------|
| **No connectivity** | Location offline | Check ActiveGate logs, network |
| **SSL errors** | Certificate failures | Install CA certs on ActiveGate |
| **Timeouts** | All tests fail | Check network routes, DNS |
| **Resource exhaustion** | Slow/failed tests | Scale ActiveGate resources |
| **Browser issues** | Clickpath failures | Check Chrome/display config |

### ActiveGate Logs

```bash
# Linux log location
/var/log/dynatrace/gateway/

# Key log files
gateway.log         # Main gateway log
synthetic.log       # Synthetic execution log
connection.log      # Cluster connectivity
```

### Network Verification

```bash
# Test connectivity to target
curl -v https://internal-api.company.com/health

# DNS resolution
nslookup internal-api.company.com

# Test Dynatrace connectivity
curl -v https://.live.dynatrace.com/api/v1/time
```

```dql
// Identify failing monitors by location
// To filter for specific private locations, add a filter on
dt.entity.synthetic_location
fetch bizevents, from: now() - 24h
| filter event.provider == "dynatrace.synthetic"
| filter synthetic.availability == false
````

```
| summarize {
    failure_count = count(),
    last_failure = max(timestamp)
  }, by: {dt.entity.synthetic_test, dt.entity.synthetic_location}
| sort failure_count desc
| limit 20
```

Summary

In this notebook, you learned:

✅ **Why private locations** - Internal apps, security, compliance
✅ **Architecture** - ActiveGate with synthetic engine
✅ **Deployment options** - Installer, container, Kubernetes
✅ **Configuration** - Creating and managing locations
✅ **Health monitoring** - ActiveGate and execution metrics
✅ **Troubleshooting** - Common issues and resolution

Next Steps

Continue to **SYNTH-05: Network Monitoring** to learn about synthetic network availability monitoring.

References

- [Private Synthetic Locations](https://docs.dynatrace.com/docs/platform-modules/digital-experience/synthetic-monitoring/private-synthetic-locations)
- [ActiveGate Deployment](https://docs.dynatrace.com/docs/setup-and-configuration/dynatrace-activegate)
- [Synthetic-Enabled ActiveGate](https://docs.dynatrace.com/docs/platform-modules/digital-experience/synthetic-monitoring/private-synthetic-locations/create-a-private-synthetic-location)
- [ActiveGate in Kubernetes](https://docs.dynatrace.com/docs/setup-and-configuration/setup-on-k8s/guides/activegate-capabilities)