Cookie Reuse and Port Availability Checks

This chapter focuses on advanced Dynatrace synthetic monitoring scenarios involving cookie persistence between monitors and checking server port availability before enabling a monitor.

24.1 Cookie Persistence Between URL Monitors

Use Case:

You have one URL sequence monitor that logs in and saves a session cookie. Another monitor needs to use this cookie to continue a related session.

Current Capabilities and Limitations:

Dynatrace **does not support cookie persistence across multiple monitors** out of the box. Each synthetic monitor execution is **stateless and independent**.

Workaround and Solution:

To achieve session sharing:

1. **Combine login and action into a single Clickpath Monitor**:
    - Step 1: Login and capture session cookie
    - Step 2: Perform the next action using the same browser context
2. **Use chained HTTP monitor requests**:
    - Step 1: POST to login endpoint and extract the session cookie/token
    - Step 2: Use extracted session value in the header or cookie of subsequent steps
3. **External orchestration (if multi-monitor chaining is required)**:
    - Use a CI/CD tool or script to:
        - Run Monitor A, extract cookie via API or logs
        - Inject that cookie into Monitor B’s HTTP headers using the Dynatrace API

This allows you to simulate session-based navigation but requires advanced configuration.

24.2 Port Availability Checks Before Monitor Creation

Scenario:

You want to verify that a target server’s port is open and listening **before creating/enabling a synthetic monitor**, similar to what SiteScope offers via basic connectivity check.

Dynatrace’s Current Behavior:

Dynatrace does not provide a **built-in port-checking utility** within the Synthetic Monitoring configuration UI. However, several options exist:

Option 1: Use External Port Scanning Tool

- Use telnet, nc (netcat), or nmap on a local or PSL-based host to verify port connectivity.

```
nc -zv yourserver.com 443
```

Option 2: Use Script Monitor (ActiveGate)

- Create a **custom script monitor** running on a Private Synthetic Location (ActiveGate) that:

```
#!/bin/bash
HOST=yourserver.com
PORT=443
if nc -zv $HOST $PORT 2>&1 | grep -q succeeded; then
  echo "Port $PORT is open"
  exit 0
else
  echo "Port $PORT is closed"
  exit 2
fi
```

- Return 0 if successful, non-zero if port is closed.
- This monitor will execute before other monitors are created/enabled.

Option 3: Passive Monitoring + Alerts

- If you already have a monitor but want to simulate port closure detection:
  - Use **HTTP Monitor** to simulate an endpoint call.
  - Alert on connection refused or timeout error codes.

Summary

Dynatrace synthetic monitors are stateless by design, which limits cookie persistence across monitors. However, session continuity can be achieved via combined steps in a single monitor or orchestration outside Dynatrace. Port checks are not built-in but can be handled effectively using custom script monitors or external tools integrated with Private Synthetic Locations.