Day - 5

14 November 2025 08:41

Today

1. Davis AI
2. DevSec
3. Hub.
4. Workflow
5. Note book.
6. Integration of Jmeter & load Jacoco
7. Instrumentation.
8. Data Retention
9. License
10. Export the config.

09-09-09
1,
Sep 09

→ Note book

1. Notebook:-

Analysis, Troubleshooting, Data exploration.

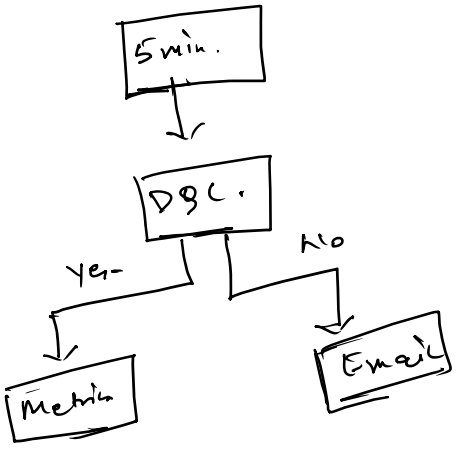
SRE, observability, Developer

1. Deep Analysis, adhoc investigation.
2. Developer, observability engineers
3. DQL,
4. Built mainly on grail + DQL.
5. Debugging & Data exploration.

Dashboard

1. End user.
2. Visualize the panel & data.

2. Workflow:-



Log level Alert

info
error > 10

Cron job → * * * * * -

* DevSecOps:-

End-to-end observability security

1. Application Security module.
2. Runtime App protection (RASP)
3. Vulnerability Detection & SCA.
   Detection & observability.

(2)
(3) Vulnerability Detection & observability
(4) Attack Detection

(1) Development phase.
(2) Runtime Phase.
(3) Prioritised & Risk based Security findings.
(4) Attack Detection & Protection.
(5) CI/CD Integration - Automation.
(6) Cloud Security Posture.
(7) Unified security

(1) Development Phase:- DT scan app. build, deploy & Runtime.

① Vulnerability Analysis → CVE Mapping
→ Version upgrade
→ Prioritised risk scoring.

② Dev Team Benefit → Dev- get immediate feedback
→ Faster Remediation.
→ Security findings integrated into CI/CD.

(2) Runtime Phase:- DT continuously monitor app in production

① Vulnerability
② Misconfiguration
③ Suspicious Behaviour
④ Exploit Attempts

(3) Prioritized & Risk Based Security findings:-

① what is Business Impact?
② Actively attacking it?
③

(4) Attack Detection & Protection:-

Actual Attack → SQL injection.
→ Command Inj.
→ Traversal.

(4) Adfeed

Actual Attack
- Command Inj.
- Directory Traversal.
- Remote code execution.

(5) CI/CD Integration + Automation

(1) Jenkins
(2) Devops
(3) GitHub Action
(4) GitHub
(5) BitBucket Pipelines.

(6) cloud Security Postures

(1) S3 Bucket.
(2) IAM Misconfig
(3) K8s
(4) Image Scan).

(7) Unified Security

(1) Dashboard - Vulnerability
Risk score.
Active Attack

Top exploited CVE

(1) Code — Library Scanning, SCA
(2) Build — Report Vulnerability
(3) Test — Runtime check
(4) Deploy — Deployment Validation, Misconfig Detection
(5) Run — RASP, exploit Detection.
(6) Monitor — Dashboard & Alert.
(7) Respond — Attack Details, RCA, Remediation Guidance.

→ DevSecOps loop.

Dynatrace Config :-

Dynatrace Config :-

1. Dashboard
2. SLO
3. Workflow
4. Notebook
5. Log ingest tables.

Tjson

Monaco
↳ monitoring as code

Dynatrace Certification

1. Dynatrace Associate Certification — 60 - MCQ, 90min, 75%
2. Dynatrace Professional Certification — 90min,
3. Dynatrace Expert certification → MCQ, Practical Exam, Interview

100%
100%
70-75%

Evening :-

1. Davis AI
2. Hub
3. Integration with Jmeter.
4. Instrumentation.

5. Dynatrace License.

(1) Davis AI :- Brain of Dynatrace.

full autonomous, causal AI-engine that automatically analyze metrics, logs, traces, events, topology, dependencies & user behaviour to detect problem, find root cause, get the precise Answer.

Understand how your system works!
↳ behaviour looks like?

① Understand how your sy...
② What normal behaviour looks like ?
③ Why something broke ?
④ Who is impacted ?

What Davis AI does?

① Automatic Anomaly Detection
② Precise Root Cause Analysis
③ Business Impact Analysis.
④ Alert Consolidation.

① Automatic Anomaly Detection:-

① Detect Unusual spikes, drops, slowdown, error bursts.
② Understand seasonal behaviour pattern.
③ Detect Anomalies in:-

① Metrics
② logs
③ events
④ Services.
⑤ Processes
⑥ Kubernetes

② Precise Root Cause Analysis:-

① Traces.
② Services
③ Dependencies.
④ logs.
⑤ Infra.
⑥ cloud component.
⑦ Network.

Root Cause → [Underlying reason that created those effects.]
Not Symptoms. → [Effect of the Problem]

③ Business Impact Analysis:-

① How many users affected?
② How many service impacted?
③ SLO impact.
④ Revenue or transaction loss.

(3) - -
(4) Revenue or transaction loss.

(4) Alert Consolidation:- Send one Problem card with full chain of impact.

Why Davis AI is Unique?

(1) Not correlation Based.
(2) Not Rule based.
(3) Not threshold Based.

Key Capabilities:-

(1) Dynamic Baselines
(2) Automatic Problem Detection.
(3) Davis RCA (Root Cause Analysis)
(4) Davis for Security (AppSec fRASP)
    (1) Attack Patter.
    (2) Exploit Attempt.
    (3) Threat actors.
    (4) zero-day Attack.

Davis Copilots:- chatgpt | copilot (LLM)

(1) Prompt

chatgpt + Dynatrace + Davis AI + Smartscape + DQL

Dynatrace licenses:-

(1) classic Host Unit Licensing (Legacy)
(2) Davis Data Unit & DEM unit (ModernModel)
- 2 | Consumption (New DT Platform | Full licen)

- (2) Davis Data Unit 4 x
- (3) Platform Based Consumption (New DT Platform | Grail Licensing)

(1.) classic Host Unit Licensing :-

(1) Dynatrace Managed
(2) DT SAAS

Host Unit :- 1 Host = 16 GB RAM = 1 HU
32 GB RAM = 2 HU

Problem?
(1) Expensive on Enviornt Scale.
(2) No Pay-as-you go.
(3) Logs charged seperately.
(4) No Grail or Copilot support.

Still used by:-
(1) old Managed cluster.
(2) Not Migrated to platform.

(2.) DEM Unit (Digital Experience Monitoring Unit) :-

used across all licensing model.
- RUM | User session.
- Session Replay.
- Mobile Monitoring
- Synthetic Monitoring

Cost Breakdown:-
1 User Session = 1 DEM Unit.
1 Synthetic HTTP Monitoring = Higher DEM usage
Session Replay = Multiple DEM Unit

1 Synthetic ...
Session Replay = Multiple DEM Unit

(3) Davis Data Unit (DDU) :-

DDU Buckets
1. Metric DDU
2. Log DDU
3. Event DDU
4. Trace DDU
5. Extension DDU

Adv :-
1. Pay for only what you ingest
2. Flexible.
3. Cloud-friendly
4. Scale easily.

Limitations :-
1. No Unified Billing.
2. No Grail.
3. Estimation consumption is hard.

(4) Dynatrace Platform Licensing (Latest, 2023-2025) :-

Platform Unit + Compute Unit + Storage Unit

Platform
↓
DQL Query.
Smartscape.
Topology
Workflows

Compute
↓
Log ingestion
Retention
Query compute.

Storage
↓
App usage Metrics

AppSec Licensing
↓
Vulnerability Detection.
Zero day Detection.
SBOM Attacks.
Attack Detection.

1. Pay to go.
2. Cheaper - Reserved one.
   1 year | 3 year
3. License Exhausted still you can use at the same price point.

(3) Integration with Jmeter:

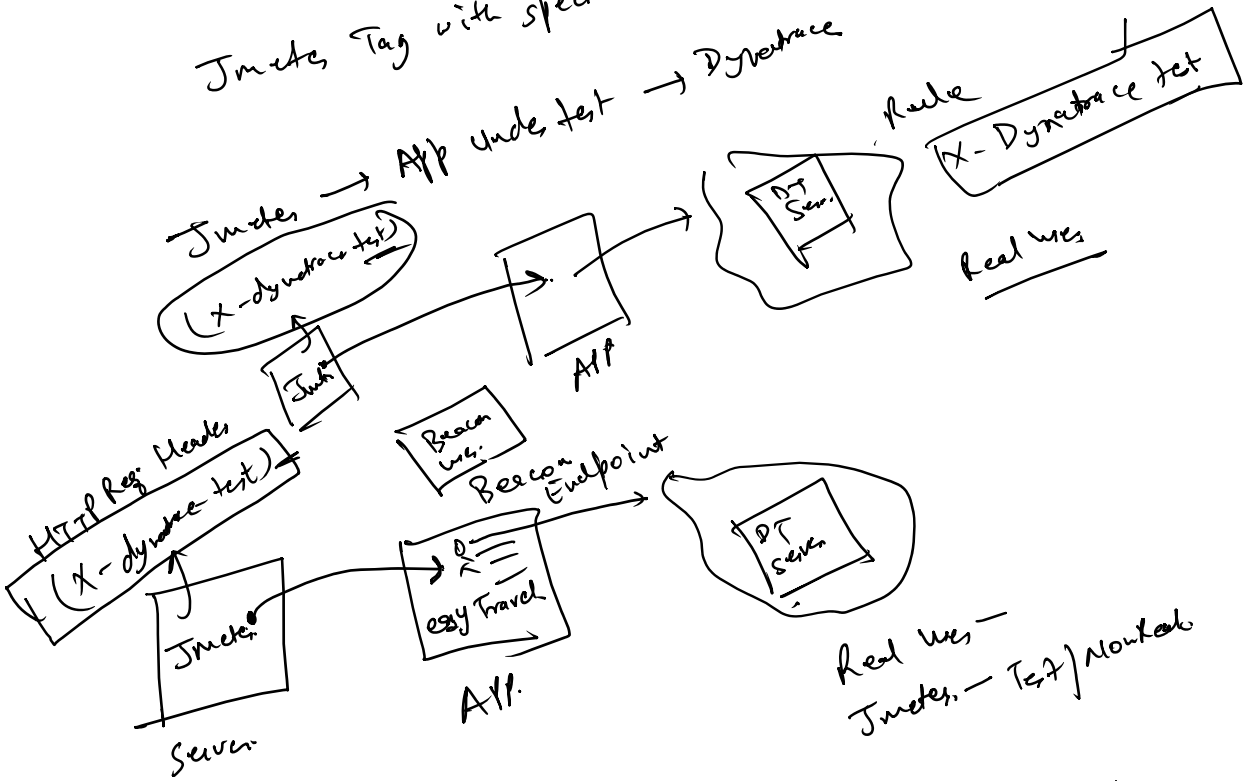
Jmeter:- Testing tool → load, stress and scalability of web App's, API's, DB & servers.

Type of test:

1. Load Testing
2. stress Testing
3. Soak Testing
4. Scalability Testing
5. Spike Testing
6. Function API's Testing

Jmeter Tag with special Header (x-dynatrace-test)

Jmeter → App under test → Dynatrace



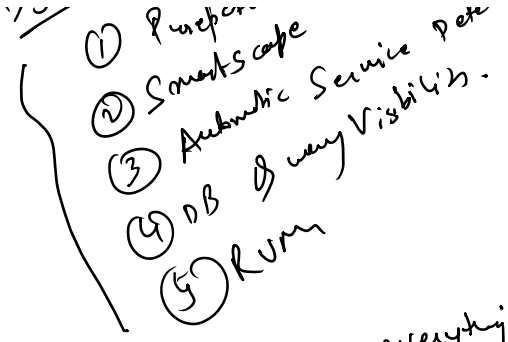
(4) Instrumentation:-

Process of automatically collecting performance, traces, logs, metrics, code level etc.

(6) cloud Monitoring

Dynatrace

1. Purepath.
2. Smartscape
3. Service Perf.
4. Visibility.



Instrumentation = How DT sees everything inside your application ecosystem?

Type of Instrumentation:—

① One Agent - Heart of DT

② Auto-instrumentation (zero touch for containers & K8s)
- Open Telemetry → One Agent
- DT operator

③ RUM Instrumentation
- end-user browser session.

④ openTelemetry (OTel + DT)
① Metrics
② Logs
③ Traces

⑤ Custom Instrumentation
- Custom service Detection Rules
- Custom method

⑥ Log Instrumentation → grail + One Agent → Dashboard

⑦ Security Instrumentation
- Vulnerability libraries
- Attack Vectors
- SQL injection
- command injection