Day - 4

13 November 2025 08:34

DPL (Dynatrace Pattern language)

log

↓

Regular expression → extract field

Dynatrace → DPL → Data.

IP address → IPV4 → IPV4 ADDR

→ ipv6

→ ipv6 ADDR.

Enum:- fixed set of named constant value.

LD - Line date.

ADDR - IP Address.

- IPV4 ADDR
- IPV6 ADDR

INT = Integer.

Timestamp

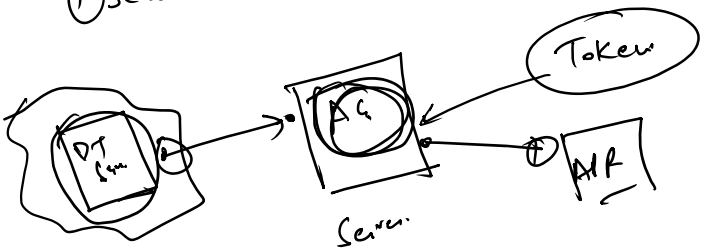
SPACE

FLOAT

ENUM → Condition

Active Gate:-

(1) Secure Communication b/w Dynatrace & source

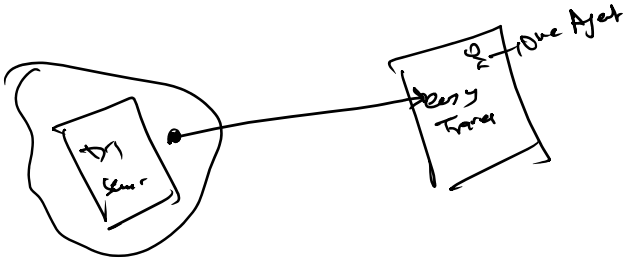


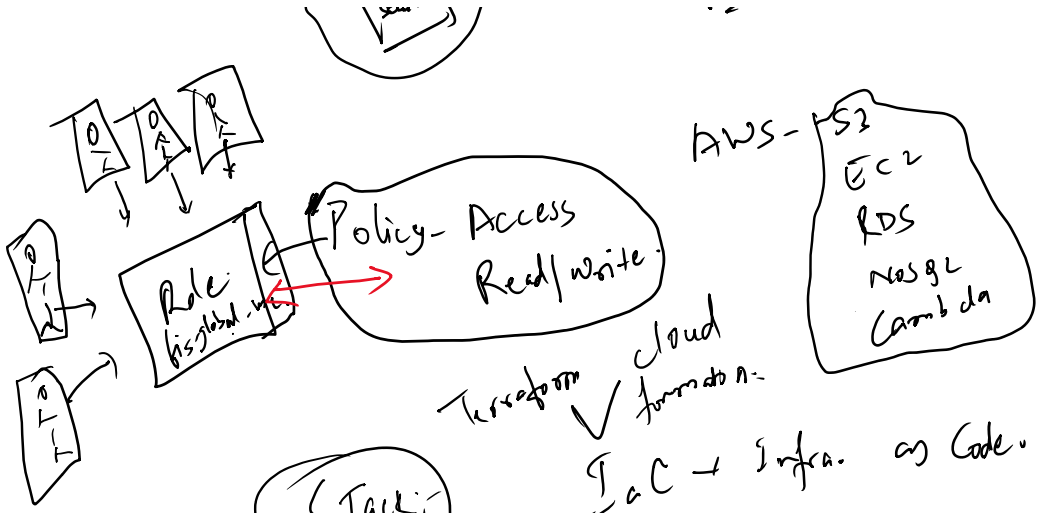
(1) Private App -

(2) cloud Monitoring

(3) DB monitoring

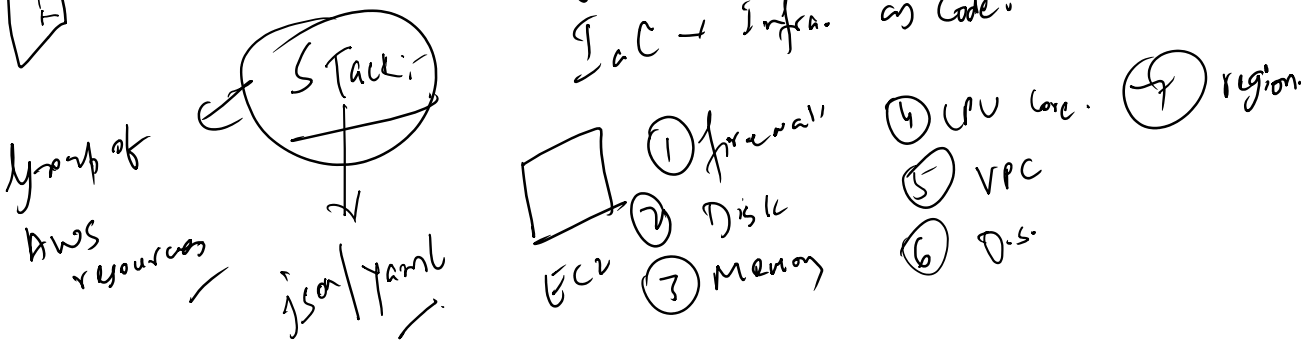
(4) DT extension.



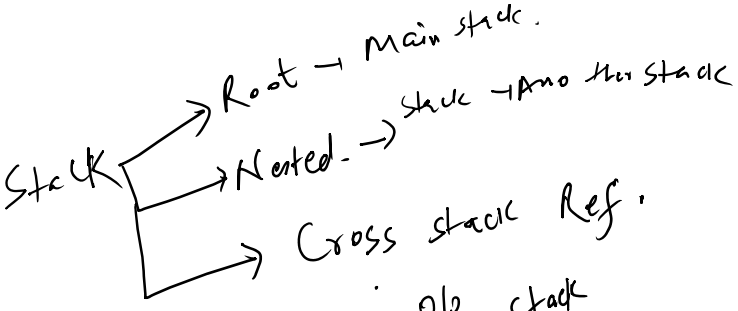


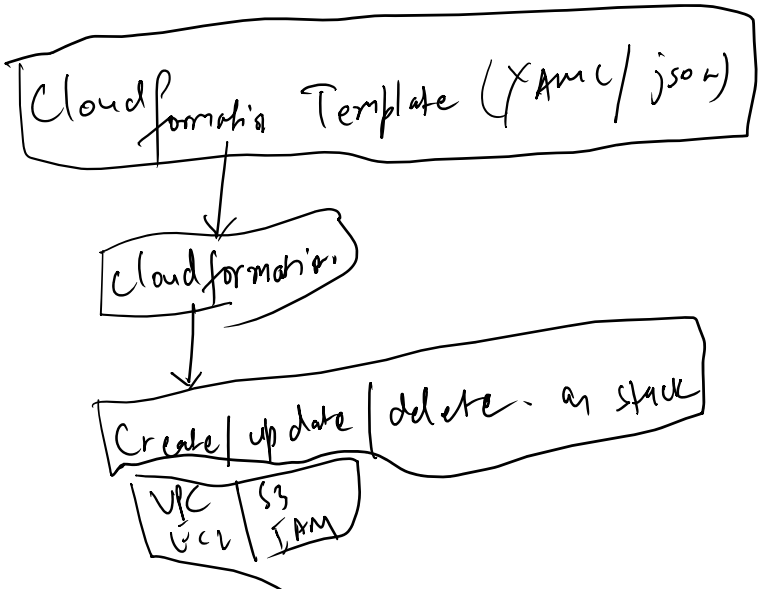
Terraform cloud formation:-

IaC → Infra. as Code.



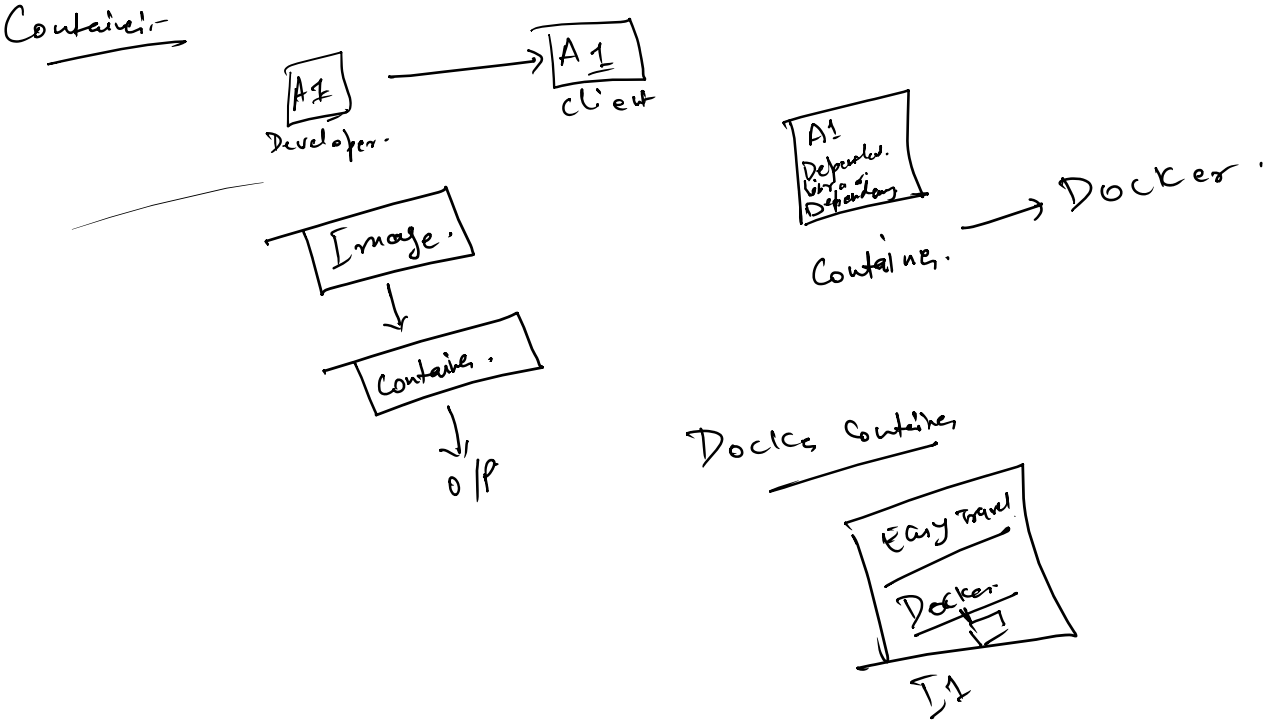
1. IaC
2. One click Deployment.
3. Consistency.
4. Easy Rollback
5. Auditable changes.





VPC | S3
EC2 | IAM

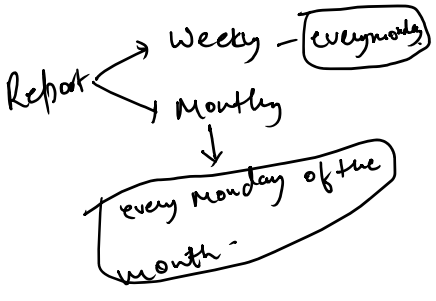
Container:-



Evening:-

1. Anamoly Detection.
2. Alert
3. Alert Integration.
4. Workflow.

1. Anamoly Detection:-



Alert - (1) Problem Alerting Profile.
(2) Vulnerability Alerting.
(3) Attack Alerting Profile.

Problem Alert (impact performance, Availability, host)

(1) Problem Alerting Profile:- Problem Alerts (impact performance) SLOs, service, host)

(1) Real-time operational Alert:
(2) Severity, impact, level, service type, m2 or problem type.
(3) Failed req.
(4) Unavailable host.
(5) Deployment changes impacting performance

(3) Platform/service monitoring.

When?
(1) On-call alert.
(2) Incident Response

(2) Vulnerability Alerting Profile:-
↳ Security weakness.

code, third party libraries runtime env.
↓
exploited by an ~~Attackers~~.

Type of Vulnerability
- 3rd party.
- Code level.
- Runtime.
- Container / Image Vulnerabilities.

(1) 3rd party Library Vulnerabilities:- (CVE Based)
↓
Common Vulnerabilities & Exposures

(1) Affected Library Version.

① Affected library Version.
② Ex- log4j vulnerabilities.
Spring4Shell
open SSL issues

② Code level Vulnerabilities :-

① Hard-code secrets
② Unsafe deserialization
③ Insecure Crypto func.
④ Outdated hashing algo.

③ Runtime env. Vulnerability :-

① Java.
② .net.
③ Insecure HTTP Headers
④ SSL certificate.

ISO-22002
ISO-20100
ISO

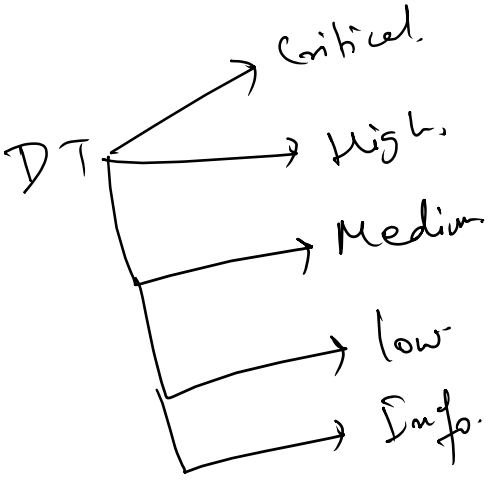
OWASP

① Security Guideline.
② Best Practices.
③ Tools.
④ Community driven standard.
⑤ Training Material

④ Container / Image Vulnerability

① OS ~~Page~~ Package
② Outdated s/w libraries.
③ libraries based containers

Image
↓
container.



Severity:-

(1) Attack Complexity.
(2) Impact Score.
(3) Public exploit presence
(4) Runtime exposure.

Used to Decide:-

(1) Which team notified
(2) severity should trigger an Alert.
(3) Tag | MZ - should be included.

Why Uniquess:-

(1) Servic - Vulnerable library
(2) Whether code is actually executed.
(3) Vulnerable is reachable.
(4) Attack surface mapping.
(5) Business impact correlation.

(3) Attack Alerting Profile (Runtime App. Protection - RASP)

RASP + Security technology that protect applications in real time, from inside the app.
during execution - Block | Monitor Malicious Behaviours

Core purpose:-

(1) Detect & Block live Attacks on the running App.
exploit Attempt

① Detect & Block
② Zero-day exploit Attempt
③ Deep Visibility into how attack interact with code.
④ stop Malicious req. based on runtime context.

① Injection Attack.

Vivek
Vivek

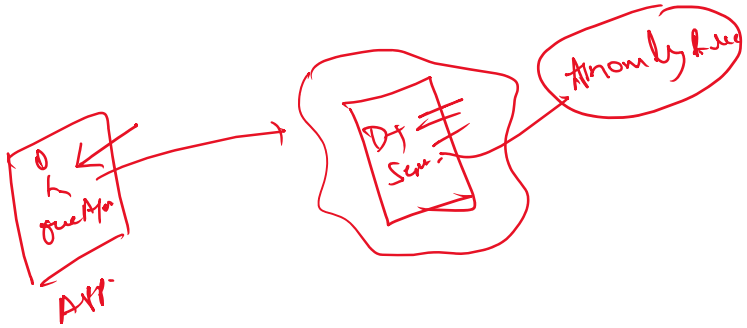
② File & Path Attack.
③ Authentication & Session Attack.
④ Advanced Runtime Exploit.
⑤ Deserialized Attack.

Dynatrace:-

① Real time attack Detection. → Active exploitation attempt across microservices.
② Attack Blocking. – Malicious req.
③ Attack correlation → App./service/library
④ MITRE Attack Mapping → Technique where attack is used
⑤ Evidence & Payload Capture.

Payload
URL
Method
IP

Parameter, Attacktype



Anomaly Detection:-

Capability that automatically identifies unusual behaviour in application, service, infra. or end user experience.

Capability that automatically identifies
your applications, service, infra. or end user experience.

AI + Baseline + Statistical Model + Deterministic logic.

↓

Detect the Problem Automatically w/o
Manual threshold.

Techniques:-

(1) Automatic Baseline Detection (Dynamic Baseline)

Historical (5-7 days) + Baseline.

(2) Davis AI Engine (Causation-Based AI)

- Doesn't generate alert storm.
- Anomaly is real.
- Customer impact.
- Downstream system are affected.
- Actual Root Cause.

(3) Deterministic Problem Detection logics

1. CPU saturation
2. Spike traffic.
3. Memory leak pattern.
4. Thread lock issues.
5. Network Degradation.
6. DB latency.
7. Synthetic failure.

Type of Anomaly Detection:-

(1) Performance Anomalies.

(2) Error Anomalies.

(1) Performance ...
(2) Error Anomalies.
(3) Infrastructure Anomalies.
(4) Availability Anomalies.
(5) RUM Anomalies
(6) Log Anomalies.

(1) Performance Anomalies:-

(1) slow Service.
(2) High latency.
(3) slow database calls.
(4) Thread Contention.

(2) Error Anomalies:-

(1) Error spike.
(2) Increased exception Rates.
(3) HTTP 4xx/5xx Anomalies

(3) Infrastructure Anomalies:-

(1) CPU usage Abnormal.
(2) Memory Saturation.
(3) Disk I/O Anomalies.
(4) Container Restarts.
(5) Node Pressure event (K8S)

(4) Availability Anomalies:-

(1) Service down.
(2) Process Crash.
(3) Synthetic Test failure.
(4) Host Unavailable.
(5) HTTP Monitor failure

(4) Host ...

(5) HTTP Monitor failure

(5) Real User Monitory Anomaly :-

1. Page load spike.
2. API latency.
3. JS error Anomaly.
4. Bounce rate surge

6. Log Anomalies :-

1. New error pattern.
2. Frequent occurance of specific error.
3. Rate Anomalies in log messages

How DT Reduce false positive!

1. Baseline specific - service, host, region, time pattern.
2. Consider Root Cause chains.
3. Smartscape to avoid alert storms.
4. Anomaly cause Customer Impact.
5. Rate of change, not just absolue threshold.

Ex:- Service - 200ms → RT
1% error rate
100 RPS Traffic.

→ 3AM
RT → 800ms
Error Rate → 15%
Traffic remain Normal.

Tomorrow:-

1. Davis AI
2. DevSec.
3. Hub.
4. Workflow.
5. Notebook