Pro Cert Study Notes

ActiveGate

- **THIS `custom.properties` overrides `config.properties`.**

Linux: `/var/lib/dynatrace/gateway/config/custom.properties`
Windows: `%ProgramData%\dynatrace\gateway\config\custom.properties`

API

- **BASE: `https://{your-environment-id}.live.dynatrace.com/api`**
- Know best practices regarding how you pass your API tokens in calls
  - Api-Token={tokenValue} at end of call
  - Don't put it in the Request URL bc it'll be visible
- Understand at a basic level how an API is used
- Dynatrace throttles/limits API calls in SaaS
  - Limits the number of API calls per second
  - Limited to 50 requests/minute
  - Synthetic endpoint limit: 1,000/request/minute
  - When limit is reached, requests return response code 429
  - Headers to check limits:
    - `X-RateLimit-Limit` --> Per-minute limit for this endpoint
    - `X-RateLimit-Remaining` --> Remaining number of requests
    - `X-RateLimit-Reset` --> Timestamp in microseconds when the limit is reset.

Data Retention

Data type	SaaS	Managed	Storage
Services: trace and code	10	Max of 365	Proprietary; Shared with non-aggregated RUM

Data type	SaaS	Managed	Storage
Services: Requests and RA	35	Max 365	Proprietary
RUM: Non-aggregated user action data	10	Max 35	Shared with distributed tracing and code insights
RUM: Aggregated user data	35	Max 35	
RUM: User Sessions	35	35	
RUM: Session Replay	Max 35	Max 35	
Synthetic	35	Max 35	
Log Monitoring	5-90	5-90	NFS based
Timseries (jey user actions and requests)	5 years	5 years	Cassandra

Timeseries

- **0-14 days:** 1 minute intervals
- **14-28 days:** 5 minute intervals
- **28-400 days**: 1 hour intervals
- **400 days - 5 years**: 1 day intervals

Data Privacy and Security

- -**IP Masking -> Settings : Preferences : Mask IP and GPS**

Data Storage

Elasticsearch

- Real User Monitoring data is is stored in `Elasticsearch` .

Cassandra

- Service level data is stored in both `transactions storage` and `Cassandra`
- `Cassandra` stores the time series data (Response times, failure rates, etc)

Transaction storage

- `Transactions storage`

Log Storage

- SaaS same AWS availability zone
- File based NFS. Common mount point throughout the cluster

Integrations

AWS

- https://www.dynatrace.com/support/help/shortlink/aws-saas-deployment#overview
- **Access method:**
  - **Key based**
  - **Role based**
- Dynatrace makes Amazon API requests every 5 minutes

Azure

- https://www.dynatrace.com/support/help/shortlink/azure-monitor-integration#my-azure-environment-is-successfully-connected-what-s-next
- Query interval is 5 minutes with a 1 minute resolution

Cloud Foundry

Google Cloud Platform

OpenShift

Kubernetes

VMware

- ActiveGate receives the data from VMware and sends it to the Dynatrace Cluster. OneAgent, which is installed on each virtual machine, provides complementary data about your infrastructure health.
- ActiveGate polls your VMware platform (vCenter or standalone ESXi hosts) to obtain information about all important resources that an ESXi server provisions to your virtual machines (for example, CPU usage, memory consumption, and datastore-related activity on your VMware platform). Dynatrace also collects information about events such as virtual machine migrations and newly created machines.
- View virtualization monitoring data

Log and Configuration Files

Configuration

Linux: `/var/lib/dynatrace/oneagent/agent/config`
Windows `C:\ProgramData\dynatrace\oneagent\agent\config`

Not to be edited (they will get overwritten)

Logs

Linux: `/opt/dynatrace/oneagent/log/`
Windows: `C:\ProgramData\dynatrace\oneagent\log`

OneAgent Configuration

Configuration file path	Use	User Modify
`deployment.conf`	Tenant Token and server settings	No
`hostautotag.conf`	Host tag config	No
`hostcustomproperties.conf`	Host custom properties	No
`infraonly.conf`	Monitoring mode set and last modified	No
`installation.conf`	Sets installation settings	No
`ruxitagentloganalytics.conf`	Log Analytics agent (masking rule)	Yes
`ruxitagentproc.conf`	AG URLs, config updates	No

Ports

Cluster Node Ports
Which network ports does ActiveGate use?

Origin	Destination	Port	Deployment
OneAgent	CAG	9999	Managed
OneAgent	EAG	9999	Both

Origin	Destination	Port	Deployment
Cluster AG	Man. Cluster Node	443	Managed
Env AG	CAG	9999	Managed
Env AG	Man. Cluster Node	443	Managed
Env AG	Cloud env	443	Both
Env AG	SaaS DT Server	443	SaaS

Alt Setup

Origin	Port	Intermediate	Port	Destination	Deployment
Env AG	9999	CAG	443	Managed Node	Managed

Log Analytics

- Log content can be filtered based on keywords or timeframe.
- You can even analyze multiple log files simultaneously—even when log files are stored across multiple hosts.
- Log monitoring enables you to create a metric based on your monitored logs
- Log content auto-discovery

Log file formats

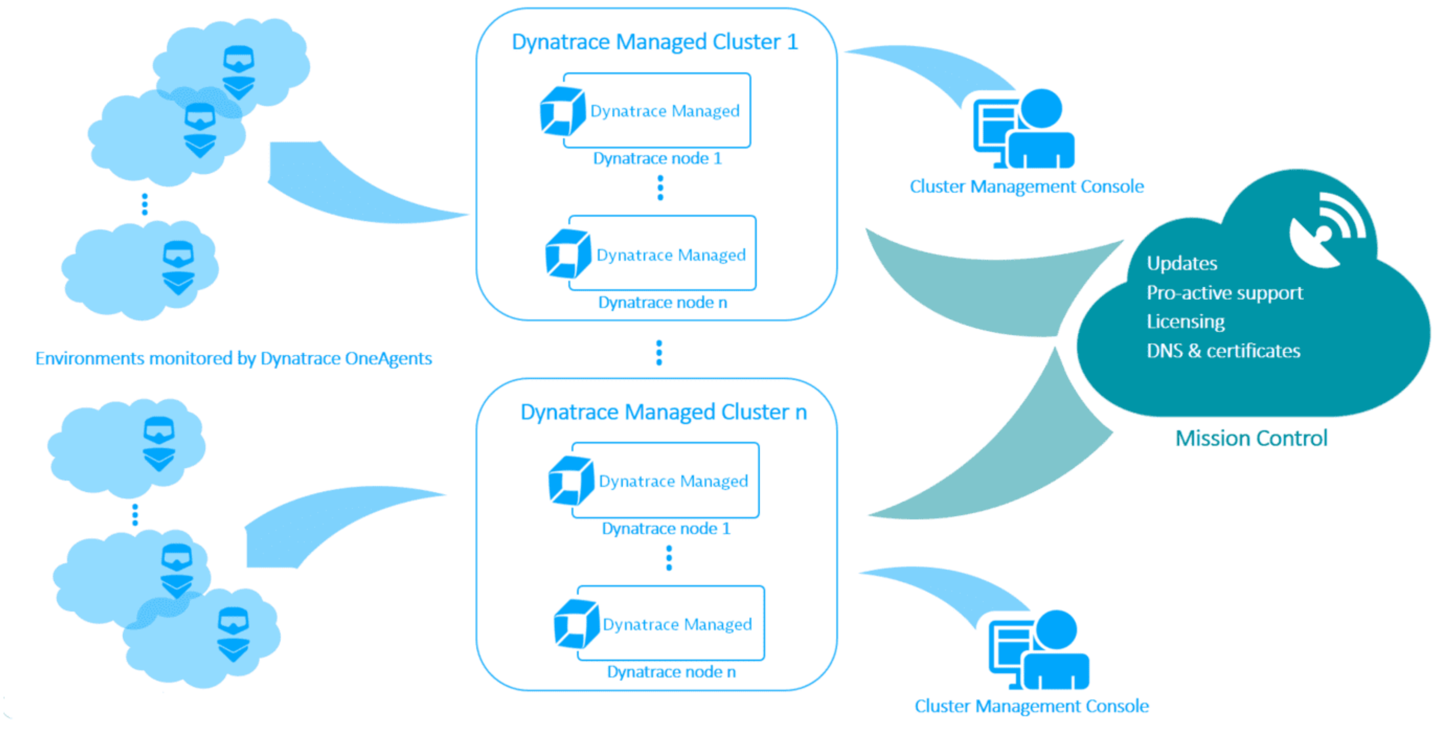
- Windows format event logs
- Plaint text file with valid timestamp
- JSON file

Log storage

- Handled in SaaS (smae AWS availability zone)
- Managed NFS mount point available to all cluster nodes

Managed

Components



- OneAgent
- Managed installer
    - **Dynatrace server**
    - **Cassandra**
    - **Elasticsearch**
    - **Embdedded ActiveGate**
    - **NGINX**
- Managed Cluster
    - At least one node
- Cluster Management Console
    - Cluster Management Console is a web-based user interface for efficiently managing your Dynatrace Managed infrastructure.
- Mission Control
    - Updates
    - Pro Active Support
    - Licensing
    - DNS and certificates

Requirements

- Dedicated host for DT Managed, host must not run other services that are CPU or memory intensive.
- 64-bit Linux distribution
- DT Server requires a fixed IP assignment
- Firewall settings that is properly configured
- Libraries installed with DT Managed are locale-aware. (in English)

Failover Mechanism

- 3 nodes per cluster
- All events, user sessions and metrics are stored with a replication factor of three.
- The entire configuration of the cluster and environments are stored on each node.
- Majority quorum is needed to run. For five nodes, two can be down.
- Latency should be less then 10 ms
- Raw transaction data (call stacks, database statements, code-level visibility) isn't replicated across nodes, it is eveenly distributed.
- In the even of a missing node, Dynatrace can accuratley estimate the missing data.

Managed Data Transfer

- https://www.dynatrace.com/support/help/shortlink/managed-data-exchange

Data type	Transfer	What is transferred	Freq
General data	Man Cluster → MC	Installation, Registration, license, health check, consumption, heartbeat, updates	
Installation data	Man Cluster → MC	Requires license key	During instaltaion and upgrade
Installation data	MC → Man Cluster	Account name, license name, OneAgent installation flag, OA download URL	During instaltaion and upgrade
Registration data	MC → Man Cluster	Reg status, username, password	Once after first startup
Registration data	Man Cluster → MC	License key, Cluster ID	Once after first startup
License data	MC → Man Cluster	License status, cluster ID, license key, license details, license model	Once every 5 minutes
Health check data	Man Cluster → MC	Cluster ID, privacy settings, time zone, traffic size, update/maintenance window, each cluster node adds: OS name and version, # of CPU cores, CPU load, total RAM, free RAM, total disk storage, used disk storage, server state, master node flag (T/F), startup time stamp. Each CAG adds: version, OS name and version, certificate issuer, type (e.g. beacon forwarder)	Once every 5 minutes
Health check data	Man Cluster → MC	Cluster ID, privacy settings, time zone, traffic size, update/maintenance window, each cluster node adds: OS name and version, # of CPU cores, CPU load, total RAM, free RAM, total disk storage, used disk storage, server state, master node flag (T/F), startup time stamp. Each CAG adds: version, OS name and version, certificate issuer, type (e.g. beacon forwarder)	Once every 5 minutes
Health check data	MC → Man Cluster	Health status, message	Once every 5 minutes
Consumption Data	Man Cluster → MC	Cluster ID, consumption timeframe, each ENV adds: # of new problems, # of RUM sessions, # of synthetic monitors. Each host adds: category, list of monitored technologies, monitored timeframes. Each synthetic monitor adds: ID, description, type, success count, failure count, action count	Once every hour
Consumption data	MC → Man Cluster	Status, remaining RUM sessions, consumed RUM overage, host units overage, remaining synthetic monitors, consumed synthetic monitors overage. For each ENV: ID, consumed RUM sessions, consumed synthetic monitors, host units.	Once every hour
Heartbeat Data	Man Cluster → MC	Cluster ID, node ID, source type (e.g. server)	Once every minute or every 5 mins if remote access disabled
Heartbeat Data	MC → Man Cluster	Remote UI request flag (T/F), websocket URLS	Once every minute or every 5 mins if remote access disabled

Data type	Transfer	What is transferred	Freq
Update data	Man Cluster → MC	Nothing	Once every hour
Update data	MC → Man Cluster	For cluster updates: version, description, download URL. For AG updates: label, version, download URLs	Once every hour

Managed User Repositories

- Manage users and groups via LDAP
- Manage users and groups with SAML in Dynatrace Managed
- Manage users and groups with OpenID in Dynatrace Managed

What happens if there is no communication from Mission Control for 14 days?

- **Email is sent out to local Admin and monitoring is turned off**

What happens during installation?

- https://www.dynatrace.com/support/help/shortlink/managed-cluster-setup#step-3-run-the-installer

The installer sets up the following components in the installation directory (by default, `/opt/dynatrace-managed`):

- Pre-configured Java Runtime Environment (your operating system settings aren't affected by this). Not visible in your alternatives options.
- Cassandra-based Hypercube storage
- Elasticsearch-based search engine
- Dynatrace Server
- An embedded ActiveGate

The installer also optimizes operating system settings:

- Swap is turned off (with `swapoff`). Note that enabling swap can result in undesired behavior and so isn't supported.
- `iptables` "PREROUTING" rules are enhanced to enable forwarding of communication to Dynatrace Server (via HTTPS on port 8021). To see the exact rules, type `iptables -L -vt nat` into your terminal.
- `readahead` page cache is set to 512.
- Limits for users are changed globally (unlimited locked-in-memory address space, unlimited address space, increased limit for number of processes and open files). See `/etc/security/limits.conf` for details.
- **`max_map_count` is modified.**

The following system files and directories may be modified during installation of Dynatrace Managed:

```
/etc/hosts
/etc/sysctl.conf
/etc/pam.d/su
/etc/rc.local
/etc/security/limits.conf
/etc/security/limits.d/90-nproc.conf
/etc/sudoers
/etc/sudoers.d/
/etc/init.d/
/etc/init.d/rc*.d/
/etc/systemd/system/
```

Cassandra Backup

- **Backup done daily, not incremental**
- Any data replicated between nodes is also stored in the backup (no deduplication)
- DT excludes the most frequently changing column families (excluded column families comprise about 80% of total storage) in addition to 1-minute and 5-minute resolution data.
- Backs up each node individually and keeps only the latest backup

Elasticsearch Backup

- Backs up entire cluster
- Data replicated across nodes and there are 2 replicas in addition to the primary shard, the backup excludes the replicated data.
- **Backup done hourly and is incremental**. Initially, DT copies the entire data set and then creates snapshots of the differences. Snapshots are copied hourly. Older snapshots are removed gradually once they are 6 months old. DT keeps at least one snapshot per month.
- For a one-node or two-node cluster, DT stores only one of the two replicas per index. Ratio of backup size to disk size is higher for one-node and two-node clusters.

Metadata

Define metadata with environment variables

- `DT_CUSTOM_PROP` can be defined on either the process or the host level
  - eg: `DT_CUSTOM_PROP=Department=Acceptance Stage=Sprint`
- `DT_TAGS` can be defined as an env variable to add a tag directly
  - eg: `DT_TAGS=MikesStuff easyTravel=Mike`

Metrics

- **Connectivity metric**: The percentage of properly established TCP conections compared to TCP connections that were refused or timed out.
  - https://www.dynatrace.com/support/help/shortlink/network-monitoring#connectivity

OneAgent

- OneAgent config
- CLI
  - `service oneagent start`
  - `systemctl start oneagent`

Installation

- https://www.dynatrace.com/support/help/shortlink/linux-custom-installation
- Overall: `--set-param=<value>`
- `--set-server`
- `--set-tenant`
- `--set-tenant-token`
- `--set-proxy`
- `--set-host-group`
- `--set-app-log-content-access`
- `--set-infra-only`
- Plus all `--set-*` parametes available via `oneagentctl`

Extensions

- OneAgent extensions requirements:
  - **Dyantrace 1.189+**
  - **OneAgent 1.189+**
  - **Python 3.6.6**
  - **OneAgent extensions SDK**
  - https://www.dynatrace.com/support/help/shortlink/oneagent-extensions-tutorial

Infrastructure monitoring

Monitoring Candidates

- https://www.dynatrace.com/support/help/reference/glossary/#expand-79monitoring-candidate

- **A host (either a VMware virtual machine or an EC2 instance) that communicates with monitored hosts in your environment, but doesn't itself have Dynatrace OneAgent installed on it. It's recommended that you install Dynatrace OneAgent on all monitoring candidates to gain full visibility and complete monitoring capabilities.**

Docker

- How Dynatrace monitors continers
- Deploy Dynatrace OneAgent as a Docker container
- The Docker metadata that is avaialble in Dynatrace

OneAgent SDK

- https://www.dynatrace.com/support/help/shortlink/oneagent-sdk

> The Dynatrace OneAgent SDK enables you to instrument your application manually to extend end-to-end visibility for frameworks and technologies for which there is no code module available yet. With the SDK, you get full access to all analysis and monitoring functionality, including auto-baselining and AI-based root cause analysis.

With the Dynatrace OneAgent SDK, you can:

- Trace incoming and outgoing remote calls
- Trace database requests
- Trace incoming and outgoing web requests
- Trace in-process asynchronous execution
- Trace queues and messages
- Capture request attributes
- User sesssion and actions are handled by OpenKit

OneAgent SDK is avaailable for:

- Java
- C/C++
- Node.js
- .NET
- Python

OpenKit

- https://www.dynatrace.com/support/help/shortlink/openkit-hub
  > With Dynatrace OpenKit, you get a set of open source libraries that enable you to instrument all other digital touchpoints in your environment, whether or not they're traditional rich client applications, smart IoT applications, or even Alexa skills.

Reference implementaitons for:

- Java
- .NET
- Native
- JavaScript

PaaS Deployments

General steps:

1. **Deploy Dynatrace --> Set up PaaS integration**
2. **Environment ID**
3. **Generate PaaS token (Settings --> Integration --> PaaS)**
4. **Needs Access problem and event feed, metrics, and topology setting enabled for API token.**
5. **Use both to integrate with PaaS**

Permissions

- https://www.dynatrace.com/support/help/shortlink/user-groups-setup#permissions

Environment Permissions

- **Access Environment**
  - Read only access
- **Change monitoring settings**
  - change env settings. Can't install OneAgent
- **View Logs**
- **View sensitive request data**
  - Allows viewing of potentially personal data. Users who don't have this permission see that the data point exists but the personal data is masked out with asterisks `(*****)`.
- **Download/Install OneAgent**
- **Configure Capture of sensitive data**
  - Allows configuration of request-attribute capture rules. These can be used to capture elements such as HTTP headers or Post parameters for storage, filtering, and search.

Account Permissions

- **Access Acount**
- **Edit billing and account info**
- **Indentify management**

Sensitive Data/Confidential Data admin

- Can view personal data (ex: method arguments) and configure request-data capture rules.

General Permission Stuff

- Users will inherit permissions from groups they are assigned
- **Test question: If you're tasked with deploying the OneAgent, which permissions do you have?**
- Deployment Admin, read only access, can't change settings.

Problems and Anomaly Detection

> A **problem** may be the result of **a single event or multiple events**, which is often the case in complex environments. To prevent a flood of seemingly unrelated problem alerts for related events in such environments, the Dynatrace AI correlates all events that share the same root cause into a single, trackable problem. This approach prevents event and alert spamming.

- the **Root Cause** may identify more than one component
- The **Impact** shows applications or components that have been affected by the problem

Sample Problem Lifespan

https://www.dynatrace.com/support/help/shortlink/problems-intro#problem-analysis

1. Dynatrace detects an infrastructure-level performance incident. A new problem is created for tracking purposes and a notification is sent out via the Dynatrace mobile app.
2. After a few minutes the infrastructure problem leads to the appearance of a performance degradation problem in one of the application's services.
3. Additional service-level performance degradation problems begin to appear. So what began as an isolated infrastructure-only problem has grown into a series of service-level problems that each have their root cause in the original incident in the infrastructure layer.
4. Eventually the service-level problems begin to affect the user experience of your customers who are interacting with your application via desktop or mobile browsers. At this point in the problem life span you have an application problem with one root cause in the infrastructure layer and additional root causes in the service layer.
5. Because Dynatrace understands all the dependencies in your environment, it correlates the performance degradation problem your customers are experiencing with the original performance problem in the infrastructure layer, thereby facilitating quick problem resolution.

Automatic Baseling

> Dynatrace checks when your applications and services are initially detected by OneAgent. The baseline cube is calculated two hours after your application or service is initially detected by OneAgent so that it can analyze two hours of actual traffic to calculate preliminary reference values and identify where your traffic comes from. Calculation of the reference cube is repeated every day so that Dynatrace can continue to adapt to changes in your traffic.

Events

> Events represent different types of individual incidents, such as metric-threshold breaches, baseline degradations, or point-in-time events, such as process crashes. Dynatrace also detects and processes informational events such as new software deployments, configuration changes, and other event types.

Event types / Severity

- **Availability**
- **Error**
- **Slowdown**
- **Resource**
- **Custom**
- **Info**

Frequent Issues

It begins with two sets of historic events:

- **Events for the last 24 hours**
- **Events for the last 7 days**

And goes as follows:

The 24-hours set is sorted in two ways:

- Duration (shortest to longest)
- Severity (less to more severe)

When a new event arrives, it is placed in proper position in each of these sorted sets.

From each initial sorted set, a subset is created, consisting of events to the right of the new event (that is, longer and more severe).

A reference set is created, consisting of events that appear in both of these two subsets.

The size of the reference set is calculated as the number of events in the reference set.

The duration of the reference set is calculated as the sum of the durations of the events in the reference set.

The same reference set is created from the 7-days set.
The following criteria are evaluated:

- **If the size of the 24-hours reference set equals or is greater than 3**, the condition is resolved as yellow. Otherwise it is resolved as red.
- If the **duration of the 24-hours reference set is equal to or greater than 50% of 24 hours** (12 hours), the condition is resolved as yellow. Otherwise it is resolved as red.
- If the **size of the 7-days reference set is equal to or greater than 7**, the condition is resolved as yellow. Otherwise it is resolved as red.
- If the **duration of the 7-days reference set is equal to or greater than 30% of 7 days** (50.4 hours), the condition is resolved as yellow. Otherwise it is resolved as red.

If at least one condition is resolved as yellow, the event is classified is yellow. Otherwise it is classified as red and an alert is triggered.

After initial evaluation, every yellow event is evaluated again with a 1-minute interval until it shifts to red or is deactivated.

Maintenance Windows

Affect on basline calculation

> Once a maintenance window is defined, Dynatrace automatically excludes the configured time period from baseline calculations. With this approach, any response time anomalies that occur during the maintenance window won't negatively influence your overall service and application baselines.

Smart Alerting

> For the generation of alerts, baselines are evaluated within **5-min and 15-min sliding time intervals**. The 5-min window serves for quick alerting in case a sufficient number of sample values surpassing a baselines are identified. A 15-min interval is used for generating alerts with higher confidence. However, in case a large amount of sample values is found to be above the baselines within one minute, Dynatrace will generate an alert at this point of time as well.
>
> To avoid over-alerting and reduce notification noise, the automated anomaly detection modes don't alert on fluctuating applications and services that haven't run for at least 20% of a full week (7 days). Alerting on response time degradations and error rate increases begins once the baseline cube is ready and the application or service has run for at least 20% of a week (7 days).
>
> Dynatrace application traffic anomaly detection is based on the assumption that most business traffic follows predictable daily and weekly traffic patterns. Dynatrace automatically learns each applications' unique traffic patterns. **Alerting on traffic spikes and drops begins after a learning period of one week because baselining requires a full week's worth of traffic** to learn daily and weekly patterns.

Visual resolution path

- The visual resolution path is an overview of the part of your topology that has been affected by the problem

Processes

Process Group Detection Rules

Types that can be detected

- IBM WebSphere clusters and domains
- Oracle WebLogic clusters and domains
- Cassandra clusters
- Tibco BusinessWorks engines
- K8s apps
- OpenShift apps
- Cloud Foundry apps
- Azure Web apps

Default process group detection rules

- **Based on Java system properties**
  - Needs to be part of the Java command line to be detected by OA
  - Java system property
- **Based on Environment Variables**
  - Covers both Java and non-Java processes like:
    - NGINX
    - Apache HTTP server
    - FPMPHP
    - Node.js
    - IIS
    - .NET
  - Can only split a PG into multiple parts. Use it if you have different deployments into the same PG.
- **Based on Process Properties**
  - Can use `DT_CLUSTER_ID` to group all processes that have the same value for this variable.
    - Needs to be set on a process-by-process basis. NOT system wide.
- To identify nodes within a process cluster that run on the same host, use `DT_NODE_ID` to tell DT which processes should be taken as separate PG instances.

Real User Monitoring

Conversion goals

- **Destination**

- **User Action**
- **Session Duration**
- **Number of User Actions**

Mobile RUM

- Analyze mobile user session crashes
- When a user session of a specific mobile user ends in a crash, you can use user session analysis to analyze the complete sequence of user actions that preceded the crash.

Session and user action properties

Expression types:

- CSS Selector
- JavaScript Variable
- Meta tag
- Cookie value
- Query string
- Server side request atttribute

Data types:

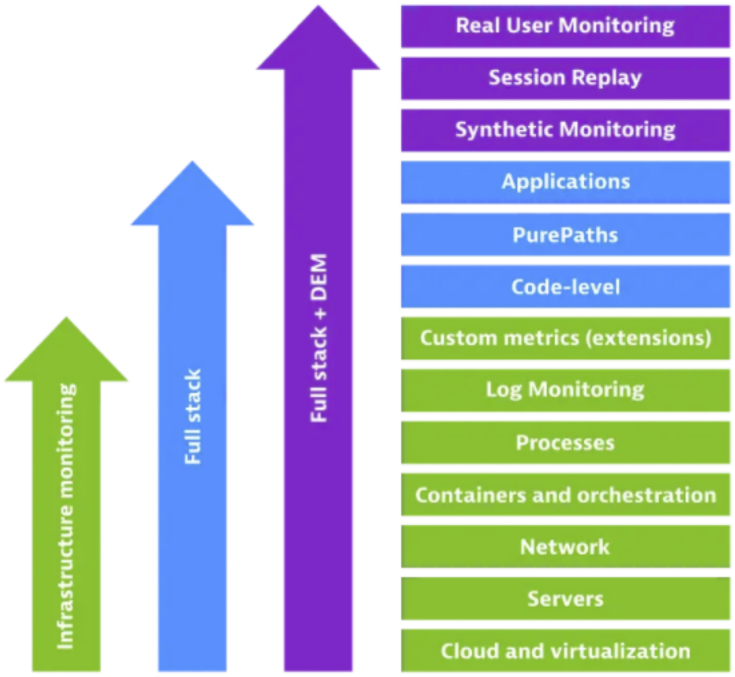
- **String**
- **Double**
- **Long**

User Actions

- User actions
- Create custom names for user actions
- Analyze individual user actions
- The User Action Dashboard tile can show action duration, actions/minute, js errors

"Visually complete" and "Speed index" metrics

- **Visually complete** is a point-in-time metric that measures when the visual area of a page has finished loading.
- Visually complete metrics are typically shorter in duration than comparable metrics (for example, page load time and DOM interactive measures) because users perceive complete page load before 100% of background page elements have loaded.



Set Action Name with `data-dtname`

```
<label for="txtFirstname">Firstname</label> <input data-dtname="Firstname Text Input" type="text" value="firstname" name="firstname" t
```

User Tag

Set up rules to automatically tag the users of this application with metadata. Expresion types that can be captured:

- **CSS Selector**
- **JavaScript Variable**
- **Meta tag**
- **Cookie value**
- **Query string**
- **Server side request atttribute**

Alternatively, you can use the JavaScript API call `dtrum.identifyUser()` to tag each user session with user-specific metadata.

Services

Define custom services

- Dynatrace allows you to define any method, class, or interface as the entry point of a service to be monitored.
- A custom service is a service that has a manually defined method, class, or interface as its entry point.
- You can define custom services for:
  - Java
  - .NET
  - PHP
  - Go
- Each custom service may contain multiple entry points.
- Entry points for your custom service can be methods of a specific class or implementations of an interface. Each non-recursive call to such a method represents a single request to your custom service.

Request attributes

- Request attributes are essentially key/value pairs that are associated with a particular service request.

- For example, if you have a travel website that tracks the destinations of each of your customers' bookings, you can set up a destination attribute for each service request. The specific value of the destination attribute of each request is populated for you automatically on all calls that include a destination attribute

Uses:

- Filtering monitoring data
- Define web-request naming rules
- Set up detection of business-logic related errors
- Build your own custom analysis charts
- Enrich PurePath analysis
- Create custom metrics
- Create custom USQL queries
- Ad hoc analysis

Can be defined by:

- **Web request data**
- **Java method arguments**
- **.NET method arguments**
- **Any data captured with OneAgent SDK**

Synthetics

- JSON is the underlying format for a script.

Still to Study

RUM questions

OA extensions/plugins
TCP connectivity errors

Practical exam

install OA,
install AG,
setup browser monitor clickpath,
application detection rules,
conversion goals,
user action naming,
key user action,
service overview,
key request,
request attributes,
request naming rules,
process group detection rules,
host group