

Internal and Confidential

Netradyne Performance Test Plan & Procedure

v2.2

Table of Contents

[Netradyne Performance Test Plan 0](https://netorg726775.sharepoint.com/sites/InfoSecDocumentGovernanceRepository/Shared%20Documents/ISO%2027001-Jul2024/ARCHIVE/CLOUD/Cloud2024/Cloud_Docs-Reviewed/Netradyne%20Performance%20Test%20Plan%20%26%20Procedure.docx#_Toc140175881)

[Document Control 2](#_Toc140175882)

[1 Purpose 3](#_Toc140175883)

[2 Scope 3](#_Toc140175884)

[3 Roles and Responsibilities 3](#_Toc140175885)

[4 Procedure 3](#_Toc140175886)

[4.1 Overview: System Architecture 3](#_Toc140175887)

[4.2 Architecture Diagram 4](#_Toc140175888)

[4.3 Detailed information on each component 4](#_Toc140175889)

[4.3.1 API Service Layer 4](#_Toc140175890)

[4.3.2 Database Layer 4](#_Toc140175891)

[4.4 Performance Test Requirements 5](#_Toc140175892)

[4.4.1 Requirements 5](#_Toc140175893)

[4.5 Performance Test Planning 5](#_Toc140175894)

[4.5.1 Performance Test Approach 5](#_Toc140175895)

[4.6 Performance Test Execution 8](#_Toc140175896)

[4.6.1 Performance Test Summary 8](#_Toc140175897)

[4.6.2 Performance Test Details 8](#_Toc140175898)

[4.6.3 Performance Test Monitoring Metrics 11](#_Toc140175899)

[4.6.4 Performance Test Environment 12](#_Toc140175900)

[4.6.5 Assumptions, Constraints, Risks and Dependencies 13](#_Toc140175901)

[4.7 Milestones 14](#_Toc140175902)

[4.7.1 Test Organization 15](#_Toc140175903)

[5 Conduct 15](#_Toc140175904)

[6 Exception 16](#_Toc140175905)

[7 Terms/Acronyms 16](#_Toc140175906)

[8 References 16](#_Toc140175907)

[9 Appendix A: Document RACI Matrix 16](#_Toc140175908)

### Document Control

|  |  |
| --- | --- |
| **Document ID** | NDCLOUDPTP2020001 |
| **Document Name** | Netradyne Performance Test Plan & Procedure |
| **Document Status** | Released |
| **Document Released Date** | 31-MAR-2020 |
| **Document Author** | Prasanna Babu |
| **Document Content Contributors** | Prasanna Babu, Tellakula Hemchand, Melwin Prabu |
| **Document Signatory** | Ankit Srivastava |
| **Document Owner** | Prasanna Babu |
| **Document Version** | v2.2 |
| **Information Classification** | Internal |

**Document Edit History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Additions/Modifications** | **Prepared/Revised By** |
| v1.0 | 31-MAR-2020 | Original Document | Prasanna Babu |
| v1.1 | 31-MAR-2021 | Annual Review | Prasanna Babu |
| v1.2 | 28-MAR-2022 | Annual Review | Prasanna Babu |
| v2.0 | 15-MAR-2023 | Aligned with standard template | Tellakula Hemchand |
| v2.1 | 01-FEB-2024 | Refined to include more controls on data privacy/Data Protection | Melwin Prabu |
| v2.2 | 27-JAN-2025 | Annual review | Prasanna Babu |

**Document Review/Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Signatory Name** | **Organization/Signatory Title** | **Comments** |
| 31-MAR-2020 | Ankit Srivastava | Senior Director |  |
| 31-MAR-2021 | Ankit Srivastava | Senior Director |  |
| 28-MAR-2022 | Ankit Srivastava | Senior Director |  |
| 15-MAR-2023 | Ankit Srivastava | Vice President |  |
| 05-FEB-2024 | Ankit Srivastava | Vice President |  |
| 29-JAN-2025 | Ankit Srivastava | Vice President |  |

**Distribution of Final Document**

|  |  |
| --- | --- |
| **Name** | **Organization/Title** |
|  |  |
|  |  |
|  |  |

# Purpose

The purpose of a performance test plan is to systematically assess the performance, scalability, and stability of a system under different conditions, identifying bottlenecks and optimizing its performance for optimal user experience and reliability.

# Scope

Netradyne provides cutting-edge technologies in AI, ML and Edge Computing to help reduce accidents by creating a new safe driving standard for commercial vehicles. Our industry solutions reduce driving incidents and protect against false claims. We empower drivers by providing them with more awareness of risky driving behaviour and reward safe driver decision-making.

In last six months the number of vehicles we are managing have been increasing exponentially. To address non-functional requirement, we have performance testing as part of our regular release cadence. The Performance test plan document will be based on which performance testing in each major release will be carried out. Performance testing has two objectives. First In each release certify release for Non-functional Requirements. Second testing readiness for future growth.

# Roles and Responsibilities

Roles and responsibilities specific to this document are included below:

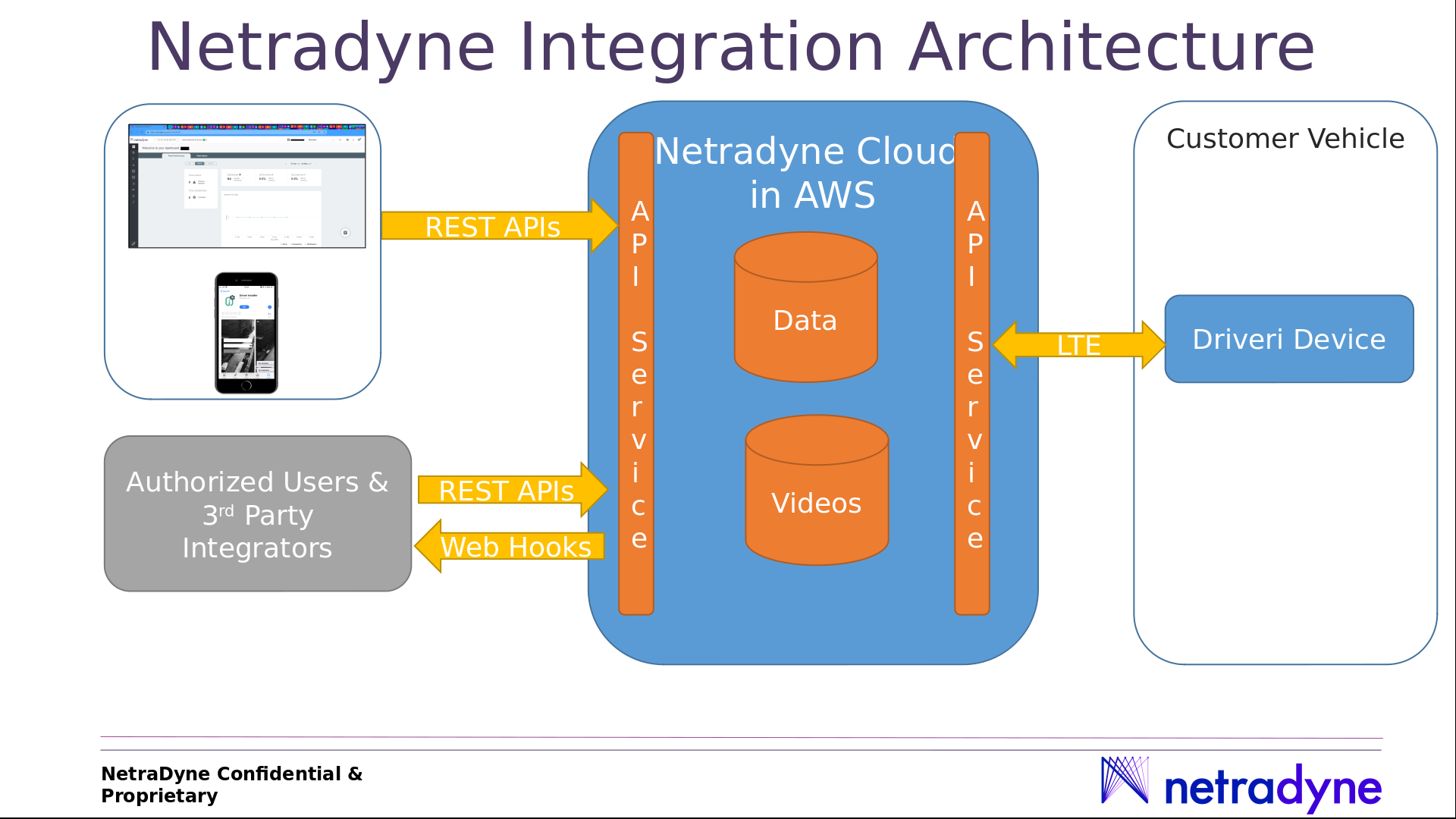
|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Owner | * Team or SME responsible for the process area needs to ensure this document is up to date and compliant with governing requirements. * Is the point of contact for the document. * Responsible for initiating and managing document review and the approval process from start to finish including gathering or delegating the collection of content including diagrams, formatting etc. as well as identifying stakeholders to participate in the peer review process. |
| Reviewers/Stakeholders | Representations from teams that can affect or be affected by the document under review (e.g., Operation, Security, Compliance, Quality) |
| Approvers | The Person(s) of authority to validate the document and sign-off on the latest version. Such Person include Document owner, Functional Team Lead, Security Lead, Product Delivery Lead. |
| Document Release | Document Owner/team to work with repository administrator to make release version available. |

# Procedure

Overview: System Architecture

The system is built on following technologies Spring, Core Java, Python. The external communication happens over rest APIs and notification to devices are sent using Amazon IOT. The Application servers are deployed in Tomcat container in most of the cases. In remaining cases applications are run as stand-alone spring/java applications. Apache web server takes the request and redirects to Applications in tomcat container. There are two load balancer s one is external facing. Other controlling traffic between internal components. Entire communication happens over HTTPS channel.

Architecture Diagram



Detailed information on each component

### API Service Layer

API Service layer handles requests from Device, Administrators Logging into IDMS Console, customer applications.

**This Layer has following main applications.**

#### App-device Service

Responsible for handling Device Related APIs like Upload Events, Upload Observations, Device Health, Upload Video etc

#### App-Console Service

Responsible for handling Administrator logging into IDMS console.

#### API-Service

Responsible for handling API s from Customer applications.

### Database Layer

Following are the PostgreSQL databases used in the system.

#### Main Database

Main database is used for managing information on tenants, admins, vehicles, devices etc.

#### Video Database

Video database used for managing information about videos.

#### Stats Database

Stats database is used for managing stats related to drivers, fleets etc.

#### Device Health Database

Device Health Database is used for managing information on device health.

#### Observation Database

Observation Database is used for managing information on observation data.

Performance Test Requirements

### Requirements

End to End Response Times Requirement for Important Business cases.

#### Business Non-Functional Requirement.

Table 3: Business NFR

|  |  |  |  |
| --- | --- | --- | --- |
|  | **90 Percentile** | **95 Percentile** | **99 Percentile** |
| **High G Event** | 5 minutes | 10 Minutes | 15 Minutes |
| **Non-High G Event** | 10 minutes | 15 Minutes | 30 Minutes |
| **Video For Events** | 30 minutes | 45 Minutes | 60 Minutes |
| **Observation Data** | 30 Minutes | 45 Minutes | 60 Minutes |
| **Live Tracking** | 2 Minute | 5 Minute | 15 Minute |
| **Web Hook Notification** | 30 Minute | 45 Minute | 60 Minute |
| **Driver Assignment** | 5 Minute | 15 Minute | 30 Minute |
| **Get Vehicle Details** | 2 Minute | 3 Minute | 5 Minute |
| **Get Driver Details** | 2 Minute | 3 Minute | 5 Minute |
| **Get Alert Details** | 2 Minute | 3 Minute | 5 Minute |

Performance Test Planning

### Performance Test Approach

Following is the high-level approach for performance testing of the application under test.

1. Understand the Functional and Non-functional Business Requirements.
   1. Input for preparing performance scripts.
   2. Input for end-to-end response times, failure rates.
   3. Input for Performance Load model.
2. Understand the Project Architecture, technologies used, databases used, High Level Design, Low Level Design and System Design.
   1. Input for Metrics to be collected.
   2. Input for Monitoring tools to be used.
   3. Identifying area of focus.
   4. Input for Performance Load Model
3. Understand the Production Load Model like concurrent device requests. Number of devices in the system, number of different API calls.
   1. Input for Performance Load Model.
4. Identify the load test type like Peak hour Load Tests, Stress Tests, Soak Test etc.
5. Identify the key performance metric areas will be important to monitor or define the pass criteria.
6. Maintain the historic performance results.
   1. Compare the current Performance run results.
   2. Input for Pass or Fail.
   3. Identifying the areas affected.
7. Predict Load for next couple of months and design load model for same.

#### Performance Testing and Monitoring Tool Details

Table 6: Description of Performance Testing and Monitoring Tool

| Tool Name | Description | Licensed / Open-Source? | No. of licenses |
| --- | --- | --- | --- |
| Locust | 1) Load Testing tool for Device APIs  2) Version used 1.4.2 | Open Source | NA |
| Apache J Meter | 1) Database Query Performance  2) Version Used 5.4.1 | Open Source | NA |
| Redis-benchmark | Used For Application Cache Performance Bench marking.  Version Used 5.0.4 | Comes with Amazon Redis cluster setup | NA |
| Telegraf | Collects CPU, Memory, Network, Disk IO, GC , Apache parameters from each Application Instance | Open Source | NA |
| Influxdb | Stores Monitoring data | Open Source | NA |
| Grafana | Dashboard for Monitoring | Open Source | NA |
| PgBadger | PostgreSQL Log analyser. Used for Database monitoring | Open Source | NA |
| Jenkins | Create Test Data  Start Performance Environment Start Performance Clients  Start Monitoring.  Start Performance Run | Open Source | NA |

#### Performance Test Script Steps

Table 7: Performance Test (Script 1 Steps)

| Step # | Device API-Service |
| --- | --- |
| 1 | Keep Alive |
| 2 | Upload Video List |
| 3 | Upload High G and Non-High G Events |
| 4 | Upload Observation File |
| 5 | Upload Video |
| 6 | Upload Health Status |
| 7 | Version Check |

Table 8: Performance Test (Script 2 Steps)

| Step # | External API-Service |
| --- | --- |
| 1 | Driver Assignment |
| 2 | Vehicle’s location |
| 3 | Alert Details |
| 4 | Get Driver Information |
| 5 | Get Vehicle details |

Table 9: Performance Test Run-time Settings

| Script # | Pacing between Iterations | Think Time between transactions |
| --- | --- | --- |
| Script 1 | 10 Minutes (Fixed) | 30 seconds (Random) |
| Script 2 | 5 Minutes (Random) | 30 Seconds (Random) |

#### Performance Test Data Planning

The Test data that will be needed for the Performance test mainly are Event Data, Observation Data, Health Data, Driver Details, Vehicles Details.

From the production database the current count as well as rate of growth of devices, tenants, drivers, vehicles ,number of requests for different Apis for last 3 months will be taken. This will be taken as input for data preparation as well as for creating Load Model.

##### **Data Preparation**

Device Health Data, Device Event Data, Observation Data , Driver Login data, New Driver details, New vehicles login data, Location data will be prepared using data prepare scripts. Which are written in Python and called from Java wrapper standalone program.

The Data preparation Jobs will be executed on Jenkins. Number of Devices, Drives, Observation file types and others are passed as parameters to the job. The Job will checkout the data prepare script on 6 Performance client machines, and it will start data prepare script on all the machines.

Post Data prepare job is done. The data is copied from Data prepare location to Data run location.

Performance Test Execution

### Performance Test Summary

The table below provides how each Performance Test script will be executed.

Table 10: Performance Test Scenarios

| Test Run | Date | Test Scenario Summary |
| --- | --- | --- |
| Smoke Test |  | To validate the performance test scripts and monitors |
| Cycle 1 - Run 1 |  | Load Test - Hour test with peak load |
| Cycle 1 - Run 2 |  | Repeat Load Test - 1 Hour test with peak load |
| Cycle 1 - Run 3 |  | Stress Test - 1 Hour test with 150% of peak load |
| Cycle 1 - Run 4 |  | Repeat Stress Test - 1 Hour test with 150% of peak load |
| Cycle 1 - Run 5 |  | Soak Test - 8 Hour Test with average load |
| Cycle 1 - Run 6 |  | Repeat Soak Test - 8 Hour Test with average load |
| Cycle 2 - Run 1 |  | Load Test - 1 Hour test with peak load |
| Cycle 2 - Run 2 |  | Repeat Load Test - 1 Hour test with peak load |
| Cycle 2 - Run 3 |  | Stress Test - 1 Hour test with 150% of peak load |
| Cycle 2 - Run 4 |  | Repeat Stress Test - 1 Hour test with 150% of peak load |
| Cycle 2 - Run 5 |  | Soak Test - 8 Hour Test with average load |
| Cycle 2 - Run 6 |  | Repeat Soak Test - 8 Hour Test with average load |

### Performance Test Details

#### Smoke Test

The smoke test is designed to ensure that the performance test scripts are working in the Performance Test Environment. The smoke test is also used for making sure the Performance Monitors that are configured for metrics collection are operating as expected.

#### Load Test

Table 11: Load Test Scenarios Detail

|  | Test Details |
| --- | --- |
| Test ID | NFT01 (Cycle 1-Run1, Cycle 1-Run2, Cycle 2-Run1 and Cycle 2 Run 1) |
| Purpose | Peak hour transaction processing will be under examination to determine if the system can maintain response times under the highest anticipated load. This test is designed to collect performance metrics on transaction throughput, response times, and system resource utilization, in comparison to Performance requirements. |
| No. of Tests | 4 (2 tests per cycle) |
| Duration | Ramp-up: 5 Minutes  Steady State: 50 Minutes  Ramp-down: 5 Minutes |
| Scripts | 1. Device API Script 2. External API Script |
| Scenario Name | Peak Hour Test |
| Covered NFR | Device and External API |
| User Load / Volume | Five thousand Devices (Threads) Load |
| Entry Criteria | 1. The code should be stable and functionally verified 2. Test Environment should be stable and ready to use 3. Test Data should be available 4. Test scripts should be ready to use 5. Smoke Test is passed |
| Exit Criteria | 1. All the NFR must be met 2. The End-To-End Response times are not 20% less than last previous three release average. 3. The error rate of transactions must not be more than 5% of total transaction count 4. CPU utilization must not be more than 60% |

#### Stress Test

Table 12: Stress Test Scenarios Detail

|  | Test Details |
| --- | --- |
| Test ID | NFT02 (Cycle 1-Run 3, Cycle 1-Run 4, Cycle 2-Run 3 and Cycle 2 Run 4) |
| Purpose | Stressing the system to see if the current system can manage workload increase for next couple of months. Also To check if it can manage sudden workload increase during some days. In this test we also check scaling vertically or horizontally to manage the load.  This test will be conducted to determine if response times can be maintained. This test is designed to collect performance metrics on transaction throughput, response times, and system resource utilization, in comparison to Performance requirements. |
| No. of Tests | 4 (2 tests per cycle) |
| Duration | Ramp-up: 5 Minutes  Steady State: 50 Minutes  Ramp-down: 5 Minutes |
| Scripts | 1. Device API Script 2. External API Script |
| Scenario Name | Stress Test Scenario |
| Covered NFR | Device and External API |
| User Load / Volume | 10000 Devices (Threads) Load |
| Entry Criteria | 1. The code should be stable and functionally verified 2. Test Environment should be stable and ready to use 3. Test Data should be available 4. Test scripts should be ready to use 5. Smoke Test is passed |
| Exit Criteria | 1. All the NFR must be met 2. The End-To-End Response times are not 20% less than last previous 3 release average. 3. The error rate of transactions must not be more than 5% of total transaction count 4. CPU utilization must not be more than 80% 5. The End-To-End Response times are not 20% less than last previous 3 release average. 6. The error rate of transactions must not be more than 5% of total transaction count |

#### Soak Test

Table 13: Soak Test Scenarios Detail

|  | Test Details |
| --- | --- |
| Test ID | NFT03 (Cycle 1-Run 5, Cycle 1-Run 6, Cycle 2-Run 5 and Cycle 2 Run 6) |
| Purpose | This soak test will determine if the system resources are recycled for re-use while processing transactions over long periods. Proper recycling of memory, CPU, and other system utilization resources is healthy for performance. This test is designed to collect performance metrics on transaction throughput, response times, and system resource utilization, in comparison to Performance requirements with o memory leakage. |
| No. of Tests | 4 (2 tests per cycle) |
| Duration | Ramp-up:30 minutes  Steady State: 7 hours  Ramp-down: 30 Minutes |
| Scripts | 1. Device API Script 2. External API Script |
| Scenario Name | Soak Test Scenario |
| Covered NFR | Device and External API |
| User Load / Volume | 5000 Devices (Threads) Load |
| Entry Criteria | 1. The code should be stable and functionally verified 2. Test Environment should be stable and ready to use 3. Test Data should be available 4. Test scripts should be ready to use 5. Smoke Test is passed |
| Exit Criteria | 1. All the NFR must be met 2. The End-To-End Response times are not 20% less than last previous 3 release average. 3. The error rate of transactions must not be more than 5% of total transaction count 4. CPU utilization must not be more than 60% 5. No Memory Leak 6. No Connection Leak 7. No Crashes. 8. Resources are releases when load is decreased. |

### Performance Test Monitoring Metrics

Table 14: Application Server Tier

| Metrics | Value Measured |
| --- | --- |
| CPU utilization | CPU utilization System, User Mode |
| Memory | Memory utilization Private, Virtual |
| Java Virtual Machine (JVM) Memory | Heap Memory, Eden Space |
| Garbage collection | GC cycles per hour |
| GC Time |
| Active Threads | Number of concurrently active threads |
| Heap size | Amount of heap allocated. |
| Memory | Memory utilization  Processes in the run queue (Procs r), User Time (CPU US), System time(CPU SV), Idle time (CPU ID), Context Switching (cs), Interrupts |
| Disk I/O | Disk I/O utilization  Read/Write per sec (r/s, w/s), Percentage busy (%b), Service Time (svc\_t) |
| Network | Collisions (Collis), Output Packets (Opkts), Input errors (Ierrs), Input Packets (Ipkts) |
| Apache Workers | Apache Busy workers and Idle workers. |

Table 15: Database Server Tier5

| Metrics | Value Measured |
| --- | --- |
| CPU utilization | CPU utilization System, User , I/O |
| Memory | Private , Virtual |
| Network | Number of Connections, Network in, Network out bytes. Packet lost, |
| Disk I/O | Disk I/O utilization  Read/Write per sec, Percentage busy (%b), Service Time (svc\_t) |
| Query | Top time-consuming queries  Highest number of times queries executed.  Index Usage. Insertion, Deletion, Select query avg response times |
| Locks | Locks duration  Number of times Locks taken  Query performance during lock times |
| Vacuum | Vacuum cycle and duration. |

### Performance Test Environment

The Performance Test environment is 10% of the production environment. Hence user load has been scaled down to 10%. Post-execution, the test result will be extrapolated with the same percentage.

As listed below, the Scaling factor between the Production environment that will support the Application under Test, and the Performance Test environment that will support the Application under Test.

The Scaling factors are as follows:

1. Number of CPUs (processors).
   1. Application Device Servers => Horizontal scaling with ratio 1: 10
   2. API servers => Horizontal scaling with ratio 1: 3
   3. Data Base => 1: 4 => Vertical Scaling [ More vCPus]
   4. App Internal => Horizontal scaling with ratio 1: 4
2. Memory
   1. Application Device Servers => 1: 1
   2. API Servers => 1: 1
   3. Data Base => 1: 4 => Vertical scaling
3. Environment configuration files => Same in Performance and as well as Production for Database, Workers, Tomcat will be same in both the environment.
4. Test Data
   1. Database => 1: 5
   2. Input Data => 1: 10

Table 16: Performance Test Environment Details

| Server Name | Tier | Amazon EC Node type | OS | Memory (GB) | vCPU | Total Disk Space |
| --- | --- | --- | --- | --- | --- | --- |
| App Device | Application Servers | m5.xlarge | Ubuntu | 16 GB | 4 cores | 200 GB |
| API | Application Servers | m5.xlarge | Ubuntu | 16 GB | 4 cores | 200 GB |
| App Internal | Application Servers | m5.xlarge | Ubuntu | 16 GB | 4 cores | 200 GB |
| Main Database | Data Base | m5.2xlarge | Ubuntu | 32 GB | 8 cores | 16 TB |
| Video Database | Database | m5.2xlarge | Ubuntu | 32 GB | 8 cores | 16 TB |
| Stats Database | Database | m5.xlarge | Ubuntu | 16 GB | 4 cores | 10 TB |
| Observation database | Database | m5.xlarge | Ubuntu | 16 GB | 4 cores | 10 TB |
| Device Health database | Database | m5.xlarge | Ubuntu | 16 GB | 4 cores | 10 TB |

### Assumptions, Constraints, Risks and Dependencies

#### Assumptions

Table 18: Assumptions

| No. | Assumption |
| --- | --- |
| 1 | The code release version is stable and passed in functional testing before deploying in the Performance Testing environment. |
| 2 | The required license must be available in the Performance Center to run the test. |
| 3 | The fully deployed, installed, and configured Web tier, middleware tier, and database servers must be operational for performance testing shake-out to begin. |
| 4 | Test Data must be provided to the performance testing team before testing starts |

#### Constraints

Table 19: Constraints

|  |  |  |
| --- | --- | --- |
| No. | Constraint | Impact |
| 1 | The Performance Test environment has 10% of the servers that Production has. | The scaling factor of the Performance Test to Production is 10%. All Production Load Models that are executed in the Performance Test should be run at 10% of the full Production load Model to represent a 100% Load Test in the Test environment. |
| 2 | The Performance Test environment will not have data older than 3 months | The database query performance in cases when large time range is selected will not be compared with production. |

#### Risks

Table 20: Risks

| No. | Risk | Impact | Action/Mitigation | Assigned To |
| --- | --- | --- | --- | --- |
| 1 | If functional errors from validation testing occur and prevent the creation of performance test scripts or performance test execution, execution of performance test project tasks will be delayed until functional errors can be addressed. | HIGH | The team will start Performance Test execution once environment certification, test script validation, and data staging efforts are completed. | Project Manager |
| 2 | If a performance-tuning effort is conducted in the middle of the performance test execution schedule and as a result configuration or code changes are made to the environment, any tests executed prior to the performance-tuning changes should be re-executed. | HIGH | It is recommended that any tests that were executed before the performance tuning changes should be re-executed after the performance-tuning changes. | Project Manager, Performance Engineering |

#### Dependencies

Table 21: Risks

| No. | Dependencies | Impact | Action/Mitigation | Assigned To |
| --- | --- | --- | --- | --- |
| 1 | The latest build should be available in the non-functional environment before NFT start date | HIGH | The team will start Performance Test execution once the environment has the latest and functionally tested code. | Development Manager. |
| 2 | Test data should be available | HIGH | Current Production count of records, devices etc should be provided to perf team so that test data can be prepared. | DBA. |

Milestones

Key milestones are listed in the table below. Each of the milestones represents a group of tasks on which completion of Performance Testing is dependent. If any of the milestones are listed as “At Risk”, the milestones that follow it will most likely be delayed as well.

Table 22: Schedule of Milestones

| ID | % Done | At Risk | Task | Due Date | Interface |
| --- | --- | --- | --- | --- | --- |
| 1 | 0-100 | Yes or No | Preliminary Project Plan submitted |  | Project Management |
| 2 | 0-100 | Yes or No | Final Project Plan submitted |  | Project Management |
| 3 | 0-100 | Yes or No | Performance Requirements and Production Load Model reviewed and verified |  | Requirements Management and Performance Engineer |
| 4 | 0-100 | Yes or No | Environment Planning |  | Environment Team and Project Management |
| 5 | 0-100 | Yes or No | Test Plan |  | Performance Engineer |
| 6 | 0-100 | Yes or No | Script Development and Data Planning |  | Performance Engineer and Vendor Project Team |
| 7 | 0-100 | Yes or No | Environment Certification and Test Script Validation |  | Project Management and Environment Team |
| 8 | 0-100 | Yes or No | Data Staging and Setup |  | Performance Engineer and Vendor Project Team |
| 9 | 0-100 | Yes or No | Performance Monitoring Configuration |  | Environment Team and Performance Engineer |
| 10 | 0-100 | Yes or No | Test Execution and Analysis |  | Performance Engineer, Monitoring Tool administrators, and Development |

### Test Organization

Table 23: Test Organization

| Name | Functional Role | Responsibilities |
| --- | --- | --- |
| Ankit Srivastava | Project Manager | Facilitating and coordinating all schedules related to SDLC phases and infrastructure |
| Prasanna Babu K | Performance Engineering Lead | Manages schedules and activities related to Performance Testing projects |
| Prasanna Babu k | Performance Engineer | Prepares for performance test execution, executes performance evaluates, analyzes performance tests, and tracks problem reports |
| Anuj Dayal | Performance Engineer | Prepares for performance test execution, executes performance tests, analyzes performance tests, and tracks problem reports |
| DevOps Team | Monitoring Support | Monitors performance tests using Performance monitors |
| Syam Krishna | Application Support | Supports performance test execution as configuration or application issues are found |
| DevOps | Performance Test Environment Support | Supports and maintains the Performance Test environment |

# Conduct

Compliance Checks to this process to be performed through various methods, including but not limited to reports, internal/external audits All tests conducted adhere to the controls outlined in both HIPAA and GDPR regulations. Additionally, our workforce receives regular Security Awareness, compliance, and role-based training opportunities. New hires undergo security awareness training within 90 days of being hired. Furthermore, employees are required to complete mandatory training, including the protection of Personally Identifiable Information (PII), Protected Health Information (PHI), and other sensitive business data, which is monitored continuously . Non-compliance will be escalated to the Netradyne leadership team.

# Exception

Exception to this procedure must be approved through the Netradyne Exception Process.

# Terms/Acronyms

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| NFR | Non-functional Requirement |
| PT | Performance Testing |
| PII | Personally Identifiable Information |
| PHI | Protected Health Information |

# References

[Netradyne Information Security Policy & Procedure.pdf](https://netorg726775.sharepoint.com/:b:/r/sites/NETRADYNEDOCUMENTMANAGEMENTPORTAL/Shared%20Documents/General/ISMS%20Published%20Documents/ISMS%202023/Netradyne%20Information%20Security%20Policy%20%26%20Procedure.pdf?csf=1&web=1&e=QDfjvy" \o "https://netorg726775.sharepoint.com/:b:/r/sites/netradynedocumentmanagementportal/shared%20documents/general/isms%20published%20documents/isms%202023/netradyne%20information%20security%20policy%20%26%20procedure.pdf?csf=1&web=1&e=qdfjvy" \t "_blank)

# Appendix A: Document RACI Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Role/Activity | Document Owner/Functional Area Lead | Document Contributor | ND Leadership | Functional Area Team | InfoSec | All ND Member(s) |
| Ensure document is kept current | A | R | I, C | R, C | C | I |
| Ensure stakeholders are kept informed | A | R | - | R | C | - |
| Ensure document contains all relevant information | A | R | I, C | R, C | C | I |
| Ensure document adheres to document governance policy | A, R | R | I | R, C | R, C | I |
| Provide SME advice | I, R | A, R | I | R, C | I, C | I |
| Gathering and adding document contents | I | A, R | I, C | R, C | C | I |
| Document Approval | A | R | I, R | I | I, R | I |

|  |  |
| --- | --- |
| Key |  |
| R | Responsible |
| A | Accountable |
| C | Consulted |
| I | Informed |