

# How to Read and Use the Scope & Fleet Analysis Tool

This tool has **three main modes**: **New Capacity Scoping**, **Fleet Analysis**, and **Node Diagnostic**. Each mode answers a different question, but they all use the same underlying logic of throughput, concurrency, and utilization.

---



## 1. New Capacity Scoping

**Purpose:** Estimate how many A-licenses (A-units) you need to support your workload.

### How to use it

1. Enter the number of tests by type (Browser, HTTP/API, Network, etc.) and their frequencies.
2. Define the average duration (seconds) of each test type.
3. Set a **Utilization Target** (e.g., 75%) and **Buffer %** (e.g., 10–15%) for safety margin.
4. Click **Calculate Required Licenses** to generate the results.

### What you'll see

- **Recommendation:** Number of A-units (licenses) and which model best fits your needs.
- **Detailed Requirement Breakdown Table:**
- **Baseline Model:** Which A-model is being evaluated (A1, A2, A3...).
- **Total Workload t/s (A1 Eq):** Total normalized test throughput (includes buffer).
- **Licenses (CPU):** Licenses required based on processing throughput.
- **Licenses (Concurrency):** Licenses required to handle simultaneous test runs.
- **Final (A-units):** The greater of CPU or Concurrency requirements.
- **Bottleneck:** Whether CPU or concurrency was the limiting factor.
- **Scheduling Detail (Pre-Buffer):** The same numbers before buffer is applied.
- **Required T/S (A1-eq):** Raw workload demand without safety margin.
- **Licenses (CPU / Concurrency):** How many licenses you'd need at bare minimum.

### How to interpret results

- **CPU-limited:** Your workload needs more processing power. Consider larger A-models.
  - **Concurrency-limited:** Too many tests overlap in time. Spread schedules or add smaller agents.
  - **Balanced:** CPU and Concurrency are close — this is an optimal configuration.
- 



## 2. Fleet Analysis

**Purpose:** Review the health, efficiency, and utilization of all nodes across your environment.

### How to use it

1. Upload your **Nodes CSV/XLSX** (or add nodes manually).

2. Optionally upload **Instances CSV/XLSX** to map CPU/memory per instance.
3. Click **Run Fleet Report**.

### What you'll see

- **Current Fleet Snapshot:** Editable table of nodes, utilization %, tests/hour, and errors/hour.
- **Fleet Health Report:** Summary of all nodes and their health status.
- **License Util:** Shows both reported and derived utilization.
- **Test Volume:** Indicates throughput (T/H) by node.
- **Error Rate:** Errors per hour; color-coded by severity.
- **Status & Recommendation:** Action guidance for each node.
- **Instance Resource Warnings:** Lists high CPU or memory utilization at instance level.

### How to interpret results

- **OK (Green):** Utilization < 80% and low error rates.
- **WARN (Yellow):** Utilization between 80–90% or moderate errors.
- **DANGER (Red):** Utilization > 90% or high errors — scale or redistribute tests.



## 3. Node Diagnostic

**Purpose:** Deep-dive into a single node's health and performance.

### How to use it

1. Select the node's **Model (A1–A6)** and active instance count.
2. Enter per-hour test counts by type and their average durations.
3. Set **Run Rate %**, **Errors/hour**, and **Node Processing Time (s)**.
4. Add instance details (cores, memory, CPU%) if known.
5. Click **Run Diagnostic**.

### What you'll see

- **Diagnostic Report:**
- **Status:** Healthy / Warning / Critical.
- **Reported Utilization:** From monitoring data.
- **Derived Utilization:** Computed from test load, durations, and proc time.
- **A1-eq Load:** Node's workload normalized to A1-equivalent throughput.
- **Capacity (t/s):** Node's total possible throughput based on model.
- **Available Headroom:** Capacity left until reaching safe utilization.
- **Warnings:** Highlights approaching saturation or excessive errors.

### How to interpret results

- **Healthy:** Node utilization is safe; errors and proc time normal.
- **Warning:** Node is near capacity ( $\geq 80\%$ ) or showing elevated errors.
- **Critical:** Node overloaded; add capacity or reduce workload.

---

## How All Three Modes Work Together

Mode	Question It Answers	Output
<b>New Capacity Scoping</b>	"How many licenses do I need?"	License recommendation by model and bottleneck type.
<b>Fleet Analysis</b>	"How healthy are my existing nodes?"	Fleet utilization, error trends, and scaling suggestions.
<b>Node Diagnostic</b>	"Is this node under/over-utilized?"	Detailed node report with per-node utilization and capacity health.

---

## Key Terms (Quick Reference)

- **A1-eq t/s**: Total tests per second normalized to A1 license capacity.
  - **Licenses (CPU)**: Licenses needed to handle average processing demand.
  - **Licenses (Concurrency)**: Licenses needed to handle overlapping tests.
  - **Final (A-units)**: The greater of the two, adjusted for buffer.
  - **Bottleneck**: The factor (CPU or concurrency) that determines the recommendation.
  - **Buffer %**: Safety margin to absorb real-world delays or load spikes.
  - **Node Proc Time (s)**: Node-level test processing duration, used for diagnostic accuracy.
- 

## TL;DR Summary

- The tool checks **CPU vs. Concurrency** for each A-model.
- The **larger value** determines your required license count.
- A **15% buffer** ensures stability in production.
- Fleet and Node Diagnostics reuse the same math to show real-time health.

This makes the tool an end-to-end capacity planner and performance monitor — from planning to diagnostics.