

Inventaire — Étape 4 : dashboard, graphes et historique

Étape 4 du projet inventaire : dashboard global du parc, KPI et graphique d'évolution avec Chart.js.

5TQ

📊 Exercice

Tu as maintenant une fiche par PC. Cette étape ajoute la **vue d'ensemble** du parc, un **graphique d'évolution** et quelques **KPI globaux**.

Objectif de l'étape

Construire :

1. Un **dashboard** (page d'accueil `/inventaire`) listant tous les PC avec leurs indicateurs clés.
2. Sur la fiche PC, un **graphique d'évolution** de l'espace disque libre au fil des rapports.
3. Quelques **indicateurs globaux** sur le parc (nombre de PC, RAM totale, espace disque cumulé, dernier rapport reçu).

Partie 1 — Lister tous les PC

Ajoute dans `src/storage.php` :

```
1 function lister_pcs(): array
2 {
3     if (!is_dir(DATA_ROOT)) {
4         return [];
5     }
6
7     $resultats = [];
8     foreach (scandir(DATA_ROOT) as $entree) {
9         if ($entree === '.' || $entree === '..') continue;
10        $fichier = DATA_ROOT . "/" . $entree . "/latest.json";
11        if (is_file($fichier)) {
12            $data = json_decode(file_get_contents($fichier), true);
13            if (is_array($data)) {
14                $resultats[] = $data;
15            }
16        }
17    }
18 }
```

```

16     }
17 }
18
19 // Tri par hostname
20 usort($resultats, fn($a, $b) => strcmp($a['hostname'] ?? '', $b['hostname'] ?? ''));
21 return $resultats;
22 }

```

💡 **SCANDIR EST VOLONTAIREMENT SIMPLE** ici. Sur un vrai parc avec 10 000 PC, on indexerait dans une vraie base. Sur notre parc-classe à 8 PC, ça marche très bien et **ça reste lisible**.

Partie 2 – Le dashboard

Crée `public/dashboard.php` :

```

<?php
1 declare(strict_types=1);
2 require __DIR__ . '/../src/storage.php';
3
4 $pcs = lister_pcs();
5
6 // Indicateurs globaux
7 $nb = count($pcs);
8 $ramTotale = 0;
9 $diskTotal = 0;
10 $diskLibre = 0;
11 $dernierRapport = null;
12
13 foreach ($pcs as $pc) {
14     $ramTotale += (float)($pc['memory']['totalGo'] ?? 0);
15     foreach (($pc['disks'] ?? []) as $d) {
16         $diskTotal += (float)($d['totalGo'] ?? 0);
17         $diskLibre += (float)($d['freeGo'] ?? 0);
18     }
19     $t = $pc['timestamp'] ?? null;
20     if ($t && (!$dernierRapport || $t > $dernierRapport)) {
21         $dernierRapport = $t;
22     }
23 }
24
25 $titre = "Dashboard";
26 require __DIR__ . '/partials/header.php';
27
28 <h1 class="h2 mb-4">Parc informatique – vue d'ensemble</h1>
29
30 <!-- KPI cards -->

```

```

31 | <div class="row g-3 mb-4">
32 |     <div class="col-md-3">
33 |         <div class="card text-center">
34 |             <div class="card-body">
35 |                 <div class="display-5"><?=$nb ?></div>
36 |                 <div class="text-secondary">PC inventoriés</div>
37 |             </div>
38 |         </div>
39 |     </div>
40 |     <div class="col-md-3">
41 |         <div class="card text-center">
42 |             <div class="card-body">
43 |                 <div class="display-5"><?=$number_format($ramTotale, 0, ',', ' ') ?></div>
44 |                 <div class="text-secondary">Go de RAM cumulés</div>
45 |             </div>
46 |         </div>
47 |     </div>
48 |     <!-- À toi : disque total, disque libre, % de rapports < 24 h -->
49 | </div>
50 |
51 | <!-- Liste des PC -->
52 | <h2 class="h5">PC</h2>
53 | <div class="row g-3">
54 | <?php foreach ($pcs as $pc):
55 |     $hostname = $pc['hostname'] ?? '?';
56 |     $ramT      = (float)$pc['memory']['totalGo'] ?? 0;
57 |     $disque0   = $pc['disks'][0] ?? null;
58 |     $usageDisk = ($disque0 && $disque0['totalGo'])
59 |         ? round((1 - $disque0['freeGo']/$disque0['totalGo']) * 100)
60 |         : 0;
61 |     <?php >
62 |     <div class="col-md-6 col-lg-4">
63 |     <a href="/inventaire/pc/<?=$urlencode($hostname) ?>"
64 |     class="card h-100 text-decoration-none">
65 |         <div class="card-body">
66 |             <h5 class="card-title"><?=$htmlspecialchars($hostname) ?></h5>
67 |             <p class="text-secondary small mb-2">
68 |                 <?=$htmlspecialchars($pc['system']['os'] ?? '') ?>
69 |             </p>
70 |             <div class="d-flex justify-content-between">
71 |                 <span>📁 <?=$ramT ?> Go</span>
72 |                 <span>🟡 <?=$usageDisk ?> % utilisé</span>
73 |             </div>
74 |         </div>
75 |     </a>
76 | </div>
77 | <?php endforeach; ?>
78 | </div>
79 |
80 | <?php require __DIR__ . '/partials/footer.php'; ?>

```

💡 Astuce : un `<a>` avec `class="card"` donne une **carte cliquable** avec toute la zone active. Très bon pour le tactile.

Partie 3 – Préparer l'historique pour le graphe

Dans `src/storage.php`, ajoute :

```
1 function charger_historique(string $hostname, int $limite = 30): array
2 {
3     $dossier = DATA_ROOT . "/$hostname/history";
4     if (!is_dir($dossier)) return [];
5
6     $fichiers = glob($dossier . '/*.json');
7     sort($fichiers); // ordre alphabétique = ordre chronologique grâce à Ymd-His
8
9     // Ne garder que les N derniers
10    $fichiers = array_slice($fichiers, -$limite);
11
12    $rapports = [];
13    foreach ($fichiers as $f) {
14        $r = json_decode(file_get_contents($f), true);
15        if (is_array($r)) $rapports[] = $r;
16    }
17    return $rapports;
18 }
```

⚠ POURQUOI YMD-HIS EST IMPORTANT ? Parce que `sort()` trie en ordre **alphabétique** sur le nom de fichier. Avec un format `Ymd-His` (ex : `20260513-103015`), l'ordre alphabétique = l'ordre chronologique. Si tu avais utilisé `13-05-2026`, ton historique serait dans le désordre.

Partie 4 – Le graphique d'évolution avec Chart.js

Sur la fiche PC (`public/pc.php`), récupère l'historique :

```
1 $historique = charger_historique($hostname, 30);
2
3 // On prépare les données pour le JS
4 $labels = [];
5 $freeDataC = [];
6 foreach ($historique as $h) {
7     $labels[] = $h['timestamp'] ?? '';
```

```

8     $diskC = null;
9     foreach (($h['disks'] ?? []) as $d) {
10        if (($d['letter'] ?? '') === 'C') { $diskC = $d; break; }
11    }
12    $freeDataC[] = $diskC ? (float)$diskC['freeGo'] : null;
13 }

```

Et ajoute dans la page un canvas + un petit script :

```

1 <section class="mb-4">
2     <h2 class="h5">Espace libre – disque C: (Go)</h2>
3     <canvas id="chartDisque" height="100"></canvas>
4 </section>
5
6 <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.0/dist/chart.umd.min.js"></script>
<script>
7     const labels = <?=json_encode($labels) ?>;
8     const freeData = <?=json_encode($freeDataC) ?>;
9
10    new Chart(document.getElementById('chartDisque'), {
11        type: 'line',
12        data: {
13            labels: labels.map(t => new Date(t).toLocaleString('fr-BE', {
14                day: '2-digit', month: '2-digit', hour: '2-digit', minute: '2-digit'
15            })),
16            datasets: [{
17                label: 'Espace libre C: (Go)',
18                data: freeData,
19                tension: 0.3,
20                fill: true,
21                backgroundColor: 'rgba(99, 102, 241, 0.15)',
22                borderColor: 'rgba(99, 102, 241, 1)',
23                pointRadius: 3
24            }]
25        },
26        options: {
27            responsive: true,
28            plugins: { legend: { display: false } },
29            scales: {
30                y: { beginAtZero: false, ticks: { color: '#9ca3af' } },
31                x: { ticks: { color: '#9ca3af' } }
32            }
33        }
34    });
</script>

```

💡 Pour tester sans attendre, **REJOUER TON SCRIPT POWERSHELL 4-5 FOIS D’AFFILÉE** en modifiant un peu les valeurs (lance un téléchargement, ferme une appli...). Tu auras 5 points sur ton graphique.

Partie 5 – Tests intermédiaires

Trois choses à vérifier :

a) Performance

Sur le dashboard, ouvre les **outils de développement** → Réseau. Le HTML doit charger en **moins de 200 ms** même avec 50 PC simulés. Si c'est plus lent : tu fais probablement de la lecture JSON inutilement en boucle. Mets un `var_dump(memory_get_peak_usage())` pour t'en rendre compte.

b) Cas vides

Renomme temporairement le dossier `data/pcs/` en `data/pcs.bak/` et recharge `/inventaire`. Tu dois voir un message « Aucun PC inventorié » **et non un crash**.

c) Mode sombre

Bascule `data-bs-theme="dark"` ↔ `light` dans `header.php`. Le rendu doit rester propre dans les deux modes. Ajuste `app.css` si besoin pour les zones non couvertes par Bootstrap (par exemple le tracé du graphe).

Partie 6 – Indicateur d'écoresponsabilité (au choix)

Ajoute **au moins l'un** des indicateurs suivants au dashboard :

- % de PC avec un disque > 85 % plein (besoin d'action).
- Moyenne de RAM utilisée sur la dernière semaine (parc sous-utilisé ?).
- Nombre d'applications inutiles potentielles (apps installées mais jamais dans le top processus de la semaine).
- Âge moyen des derniers rapports (parc bien suivi ou abandonné).



Travailler avec l'IA – pour choisir une visualisation

Bon prompt :

J'ai pour chaque PC une liste de rapports horodatés contenant `disks[i].freeGo`. Je veux **un seul indicateur visuel** pour le dashboard, qui montre d'un coup d'œil les PC dont l'espace disque diminue rapidement. Propose-moi 3 façons de visualiser ça avec Chart.js, avec leurs avantages et inconvénients (pas de code). Je choisirai après.

L'IA propose : sparklines dans chaque carte, heatmap, classement, bar chart différentiel... **Tu choisis**. Ensuite tu re-prompts pour implémenter celle que tu as retenue.

C'est la séquence : **explorer** → **choisir** → **implémenter**. Trois prompts, pas un.

✓ Critères de réussite

- Le dashboard montre la liste des PC et au moins 3 KPI.
- Les cartes sont cliquables et mènent à la fiche PC correspondante.
- Le graphique Chart.js s'affiche correctement (axes lisibles).
- Le graphique se met à jour quand tu envoies un nouveau rapport.
- Tu as au moins un indicateur d'écoresponsabilité.
- Aucun crash sur un parc vide.

⚠ Pièges fréquents

Piège	Solution
Chart.js ne s'affiche pas	Vérifier le <code><canvas></code> dans le DOM avant le <code><script></code>
Les dates s'affichent en anglais	<code>toLocaleString('fr-BE', ...)</code>
<code>glob()</code> renvoie <code>false</code>	Le dossier n'existe pas — <code>is_dir</code> d'abord
Historique dans le désordre	Format <code>Ymd-His</code> impératif (ou tri sur <code>timestamp</code>)
Le dashboard rame avec beaucoup de PC	Ne lire que <code>latest.json</code> , pas tout l'historique

Suite

À l'étape 5, on ajoute **ton module spécifique** — celui qui fait que ton PC se distingue dans le parc. Réseau, sécurité, USB, etc., selon ton numéro.