

Xandeum pNode Atlas

Live analytics for the Xandeum pNode network. The dashboard ingests gossip data directly from pNodes via pRPC ([get-pods-with-stats](#)) and visualises health, storage utilisation, release adoption, and node-level telemetry.

Features

- Seed-aware pRPC client with automatic retry + caching
- Aggregated metrics (uptime, storage, status mix, release histogram)
- Searchable/filterable node directory with CSV/JSON export
- Version distribution + storage band insights + top storage providers
- Auto-refresh (30s) with manual override
- In-app seed override panel to target any custom gossip scouts
- Slide-over inspector per node with raw JSON export
- Rolling history capture with trend chart (stored locally under [data/pnode-history.json](#))

Getting Started

```
pnpm install  
pnpm dev
```

Visit <http://localhost:3000> to view the dashboard.

Environment

Optional [.env.local](#) (values shown are defaults):

```
PNODE_SEEDS=173.212.220.65,161.97.97.41,192.190.136.36,192.190.136.38,20  
7.244.255.1,192.190.136.28,192.190.136.29,173.212.203.145  
PNODE_RPC_PORT=6000  
PNODE_CACHE_TTL=25000          # ms  
PNODE_REQUEST_TIMEOUT=7000      # ms  
PNODE_STALE_SECONDS=1800        # mark nodes stale after 30 min  
PNODE_HISTORY_LIMIT=288         # max snapshots to persist for trends
```

You can point the app at custom seeds (IP or hostnames) either via the [.env](#) above or through the "Discovery list" control in the UI (comma-separated). The server-side fetcher will try [get-pods-with-stats](#), fall back to [get-pods](#), deduplicate nodes by pubkey, and cache the snapshot for [PNODE_CACHE_TTL](#) ms per discovery list.

Scripts

- [pnpm dev](#) – run the Next.js dev server
- [pnpm build && pnpm start](#) – production build + serve

- `pnpm lint` – lint the codebase

Deployment

Any Node 18+ target works (Vercel, Netlify, self-hosted). Ensure outbound access to your chosen pNode seeds over HTTP/6000 and configure the environment variables above. Historical data is persisted to `data/pnode-history.json`; if you deploy to a stateless host, point that path to a writable volume or replace it with an external store.