Server Health and Federation API Integrity Testing Summary - Post EC2 Federation Sync Setup

Status Report `Jun 16, 2025`

✅ Phase 1: Server Health & Port Binding

Test 1.1 – Registry EC2 Port Binding

> Confirms Node.js app is actually running and listening on `PORT=3000`.

```
sudo lsof -iTCP:3000 -sTCP:LISTEN -n -P
```

✅ **Expected**: Output showing a `node` process bound to port `3000`.

Test 1.2 – Registry Health Endpoint

> Confirms the Express server responds locally.

```
curl -i http://localhost:3000/health
```

✅ **Expected**: `HTTP/1.1 200 OK` and body: `OK`

Test 1.3 – Registry Remote Access

> Confirms the port is externally reachable from backend or browser.

```
curl -i http://18.234.49.114:3000/federation/users
```

✅ **Expected**: Valid JSON or empty `[]`

Test 1.4 – PM2 Runtime Check

> Ensures Node.js didn’t silently crash.

```
pm2 list pm2 logs registry-backend --lines 50
```

✅ **Expected**:

- Status: `online`
- No `SyntaxError`, `ReferenceError`, or port binding failures.

✅ Phase 2: Federation API Integrity

Test 2.1 – POST Federated User from Backend Node

> Sends a test federation user to registry node.

```
curl -X POST http://18.234.49.114:3000/federation/users \   -H "Content-Type: application/json" \   -H "x-api-key: secret-key-1234" \   -d '{"name":"Test User","email":"test@agrinet.org","location":"Remote","role":"observer"}'
```

✅ **Expected**: `200 OK` or `User already federated`

Test 2.2 – GET All Federated Users

> Confirms the endpoint returns real data from MongoDB.

```
curl -i http://18.234.49.114:3000/federation/users
```

✅ **Expected**: Array of users like `[{ "name": ..., "email": ... }]`

✅ Phase 3: Wix Frontend → Backend Integration

Test 3.1 – Registry Health via Wix API Domain

> From browser or Postman, visit:

```
https://registry.ntari.org/health
```

✅ **Expected**: `200 OK` plain text response

Test 3.2 – Frontend Call to Registry Federation

> From browser DevTools (Network tab), confirm call to:

```
fetch("https://registry.ntari.org/federation/users")
```

✅ **Expected**: Returns federated user list
⚠️ Must ensure:

- CORS is configured with `credentials: true`
- Origin: `https://www.ntari.org`

✅ Phase 4: Backend Node System Check (api.ntari.org)

Test 4.1 – Backend Health Check

```
curl -i https://api.ntari.org/health
```

✅ **Expected**: `200 OK`

Test 4.2 – Federation Sync Script (Backend → Registry)

> Run sync manually or check PM2 log:

```
node federation-sync.js
```

✅ **Expected**:

- `🛸 Federation Sync Job Started`
- `✅ Synced user [x]`
- `📬 Job Completed Successfully`

✅ Phase 5: Federation System-Wide Connectivity

Test 5.1 – DNS Resolution from Backend to Registry

> Confirm backend EC2 can resolve and connect to registry.

```
curl -i https://registry.ntari.org/health
```

✅ **Expected**: No DNS, SSL, or timeout error

Test 5.2 – HTTPS Check from External Browser

> Access this from any external device (e.g. phone):

```
https://registry.ntari.org/federation/users
```

✅ **Expected**: Live response from Express app over SSL

✅ Summary Matrix

Test #	Scope	Purpose	Status
1.1	Registry Node	Server listening on port	✅
1.2	Registry Node	Health route local check	✅
1.3	Backend → Registry	External federation route access	✅
2.1	Federation POST	Registry accepts new user	✅
2.2	Federation GET	Registry returns user list	✅
3.1	Wix → Registry	Wix API domain is live	✅
3.2	Wix → Federation Route	Fetch call returns expected result	✅

4.1	api.ntari.org	Backend health check	✅
4.2	Sync Script	Federation job pushes users	✅
5.1	DNS + TLS	registry.ntari.org resolves properly	✅