Integrating AWS EC2 with Wix for API Access and Custom Elements

To connect **AWS EC2 instances (e.g., `registry-node` and `backend-node`)** to the **Wix website**

🚲 NTARI.org | Network Theory Applied Research Institute and its subpaths or subdomains, this integration needs to use **Wix's external integrations** with a mix of **custom elements, HTTP functions, and proxying through API gateways or custom domains**.

🔁 Connection Strategy Overview

Goal	Tool / Method
Call EC2-hosted APIs from Wix	**Wix HTTP Functions** or **fetch() in Velo**
Display API data on Wix pages	**Custom Elements** with `<iframe>` or JS
Trigger EC2 backend actions	**Wix Velo** `fetch()` or Webhooks
Use custom subdomains or API routes	**Wix + Reverse Proxy / Domain Forwarding**

🛠️ Step-by-Step Integration

✅ 1. Expose EC2 APIs Securely

Make sure EC2 instances:

- Use HTTPS (install SSL via Let's Encrypt or use Cloudflare)
- Have public DNS or static IPs (Elastic IPs recommended)
- Use a load balancer if scaling is needed

Example:

```
http://api.ntari.org/health http://registry.ntari.org/sync
```

✅ 2. Integrate into Wix using Velo Backend

Use `fetch()` from Wix Velo backend code to call EC2 endpoints.

Example:

```jsw
// backend/federation.jsw
import { fetch } from 'wix-fetch';

export async function getNodeHealth() {
  const response = await fetch("https://api.ntari.org/health", {
    method: 'GET',
    headers: { "Content-Type": "application/json" }
  });

```

```
  if (response.ok) {
    return await response.text();
  } else {
    throw new Error(`Failed to fetch: ${response.status}`);
  }
}
```

Then call this from any frontend page or router:

```
import { getNodeHealth } from 'backend/federation.jsw';

$w.onReady(() => {
  getNodeHealth().then(status => {
    $w("#statusText").text = `API Node is: ${status}`;
  });
});
```

✅ 3. Custom Elements on Specific Pages

If trying to connect an EC2-backed dashboard or form to pages like `/network-portal`, use a **Custom Element (HTML iFrame)**:

Example:

```
<iframe src="https://api.ntari.org/portal-ui" width="100%" height="600px"></iframe>
```

This embeds your EC2 app into the Wix layout.

✅ 4. Connect to Specific Wix URLs or Routes

Wix does **not allow raw path proxying** (like turning `ntari.org/registry-node` into a true reverse proxy). But you have 3 workarounds:

Option A: Use Velo Routers

Create `router.js` files in Wix to act like middleware:

```
// routers/myRouter.js
export function myRouter_Router(request) {
  return fetch("https://registry.ntari.org/sync").then(resp =>
    resp.text().then(text =>
      ok(text, { headers: { "Content-Type": "application/json" } })
    )
  );
}
```

Attach `/network-produce`, `/network-portal`, etc., to this router in Wix site routing settings.

Option B: Use Cloudflare Workers or AWS API Gateway

- Point `registry.ntari.org` to the EC2 instance via DNS
- Use a reverse proxy worker or gateway to route `ntari.org/agrinet` to the EC2 APIs

Option C: Use `<iframe>` with authentication or token-based validation

Embed EC2-hosted pages or status widgets within a page like `https://www.ntari.org/agrinet`.

🧩 Examples of Wix Page + EC2 API Use

Wix Page URL	Embedded or Connected Feature
`/network-portal`	iFrame to EC2 dashboard with dialog status
`/fruitful-on-platform`	API call to get marketplace listings
`/pingmarket`	JSON `fetch()` to get PING data from EC2
`/sell-produce`	Post form data to backend node
`/product-page/strawberry-contract`	Contract POST → backend + response display

🔐 Bonus: Add API Key or Signature Auth

To avoid exposing EC2 endpoints publicly:

- Add token headers in `fetch()` calls
- Use request signing (e.g., HMAC or JWT)
- Require keys for `/register-node`, `/sync`, etc.