

- EC2 Instance Created** EC2 instance "MyInstance" successfully launched in the us-west-2b availability zone. The instance is running on a t2.micro instance type (free tier eligible) with status checks passed. The key pair "MyKeyPair.pem" was created for SSH authentication.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with sections like Dashboard, AWS Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, and Network & Security. The main content area has a header 'Instances (1) Info' with a search bar and filters for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Pub. Below the header is a table with one row for 'MyInstance'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Pub. The 'MyInstance' row shows 'Running', 't2.micro', '2/2 checks passed', 'us-west-2b', and 'ec2-'. At the bottom of the main area, it says 'Select an instance'.

- SSH Connection Established** Successfully connected to the EC2 instance via SSH. The command `ssh -i "MyKeyPair.pem"`

`ec2-user@ec2-44-249-32-109.us-west-2.compute.amazonaws.com` authenticates using the private key and logs into the Amazon Linux 2023 server. The fingerprint was verified and added to known hosts.

```
Last login: Mon Jan 19 15:10:45 on ttys000
[(base)] arjunavadhani@Arjuns-MacBook-Air ~ % cd Downloads
[(base)] arjunavadhani@Arjuns-MacBook-Air Downloads % chmod 400 MyKeyPair.pem
[(base)] arjunavadhani@Arjuns-MacBook-Air Downloads % ssh -i "MyKeyPair.pem" ec2-user@ec2-44-249-32-109.us-west-2.compute.amazonaws.com
The authenticity of host 'ec2-44-249-32-109.us-west-2.compute.amazonaws.com (44.249.32.109)' can't be established.
ED25519 key fingerprint is SHA256:7BDpfNy3nQ8SurNQQR/sdY7WiC0DtxjR3g6WAW7vE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-44-249-32-109.us-west-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_\_ #####          Amazon Linux 2023
~~ \####\_
~~ \#\#
~~  /\_ _   https://aws.amazon.com/linux/amazon-linux-2023
~~ V' '-' /
~~ .-' /-
~/m/
[ec2-user@ip-172-31-47-122 ~]$
```

- Uploading Binary to EC2 via SCP** Used the `scp` (secure copy) command to transfer the compiled Go binary "newmain" (27MB) from my local machine to the EC2 instance. The command authenticates using the same private key as SSH and securely copies the file to `/home/ec2-user/` on the remote server. Transfer completed at 9.4MB/s in 2 seconds.

```
(base) arjunavadhani@Arjuns-MacBook-Air ~ % scp -i /Users/arjunavadhani/Downloads/MyKeyPair.pem /Users/arjunavadhani/Documents/BSDS-course/HW1/newmain ec2-user@44.249.32.109:/home/ec2-user/
The authenticity of host '44.249.32.109 (44.249.32.109)' can't be established.
ED25519 key fingerprint is SHA256:uVYUwvDoxjRsgewAW7VE.
This host key is known by the following other names/addresses:
  ~/ssh/known_hosts:: ec2-44-249-32-109.us-west-2.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.249.32.109' (ED25519) to the list of known hosts.
newmain    100%   27MB   9.4MB/s   00:02
(base) arjunavadhani@Arjuns-MacBook-Air ~ %
```

4. Go Server Running The Go server (using the Gin framework) is running on the EC2 instance and listening on port 8080. It exposes three endpoints: GET /albums, GET /albums/:id, and POST /albums. The server successfully received a GET request to /albums and returned a 200 status code in 72.758µs.

```
[ec2-user@ip-172-31-47-122 ~]$ chmod -R 777 my_dir
[ec2-user@ip-172-31-47-122 ~]$ ./newmain
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

[GIN-debug] GET     /albums           --> main.getAlbums (3 handlers)
[GIN-debug] GET     /albums/:id        --> main.getAlbumByID (3 handlers)
[GIN-debug] POST    /albums           --> main.postAlbums (3 handlers)
[GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. We recommend you to set a value.
Please check https://github.com/gin-gonic/gin/blob/master/docs/doc.md#dont-trust-all-proxies for details.
[GIN-debug] Listening and serving HTTP on 0.0.0.0:8080
[GIN] 2026/01/19 - 21:33:37 | 200 | 72.758µs | 155.33.133.38 | GET      "/albums"
```

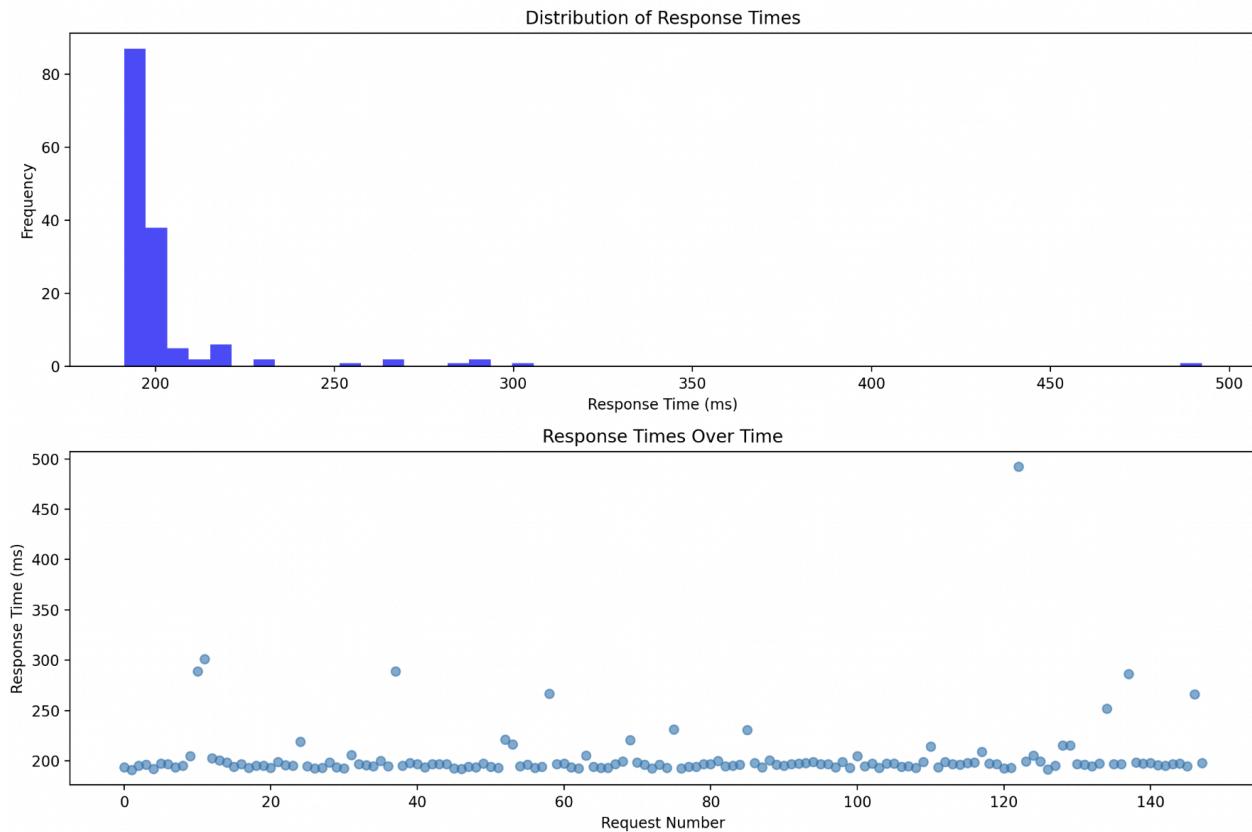
5. Successful curl Request A curl request from my local machine to the EC2 server at <http://44.249.32.109:8080/albums> returned HTTP 200 OK. The response contains JSON data with three album records (Blue Train by John Coltrane, Jeru by Gerry Mulligan, and Sarah Vaughan and Clifford Brown).

```
Last login: Mon Jan 19 16:21:24 on ttys003
[(base) arjunavadhani@Arjuns-MacBook-Air ~ % curl -v http://44.249.32.109:8080/albums
*   Trying 44.249.32.109:8080...
* Connected to 44.249.32.109 (44.249.32.109) port 8080
> GET /albums HTTP/1.1
> Host: 44.249.32.109:8080
> User-Agent: curl/8.9.1
> Accept: */*
>
* Request completely sent off
< HTTP/1.1 200 OK
< Content-Type: application/json; charset=utf-8
< Date: Mon, 19 Jan 2026 21:33:37 GMT
< Content-Length: 382
<
[
  {
    "id": "1",
    "title": "Blue Train",
    "artist": "John Coltrane",
    "price": 56.99
  },
  {
    "id": "2",
    "title": "Jeru",
    "artist": "Gerry Mulligan",
    "price": 17.99
  },
  {
    "id": "3",
    "title": "Sarah Vaughan and Clifford Brown",
    "artist": "Sarah Vaughan",
    "price": 39.99
  }
* Connection #0 to host 44.249.32.109 left intact
(base) arjunavadhani@Arjuns-MacBook-Air ~ %
```

6. Load Test Running The Python load testing script running for 30 seconds, sending continuous GET requests to the EC2 server. Each request logs its response time in milliseconds. Most requests complete in 193-200ms with occasional spikes (e.g., Request 12: 301ms, Request 38: 289ms).

```
(base) arjunavadhani@Arjuns-MacBook-Air HW2 % python load_test.py
Starting load test for 30 seconds...
Request 1: 193.88ms
Request 2: 191.23ms
Request 3: 195.37ms
Request 4: 196.69ms
Request 5: 192.26ms
Request 6: 197.39ms
Request 7: 196.84ms
Request 8: 193.65ms
Request 9: 195.26ms
Request 10: 204.90ms
Request 11: 289.01ms
Request 12: 301.17ms
Request 13: 203.13ms
Request 14: 200.57ms
Request 15: 198.38ms
Request 16: 194.21ms
Request 17: 197.11ms
Request 18: 193.25ms
Request 19: 195.39ms
Request 20: 195.26ms
Request 21: 193.57ms
Request 22: 199.18ms
Request 23: 196.17ms
Request 24: 195.35ms
Request 25: 219.39ms
Request 26: 194.99ms
Request 27: 193.05ms
Request 28: 193.56ms
Request 29: 198.38ms
Request 30: 193.86ms
Request 31: 193.02ms
Request 32: 205.92ms
Request 33: 197.30ms
Request 34: 195.97ms
Request 35: 195.07ms
Request 36: 200.09ms
Request 37: 195.14ms
Request 38: 289.34ms
Request 39: 195.50ms
Request 40: 198.32ms
Request 41: 197.28ms
Request 42: 193.88ms
Request 43: 196.82ms
Request 44: 197.24ms
Request 45: 196.97ms
Request 46: 192.92ms
Request 47: 192.18ms
Request 48: 194.53ms
Request 49: 193.79ms
Request 50: 197.72ms
Request 51: 194.51ms
Request 52: 193.51ms
Request 53: 221.18ms
Request 54: 216.62ms
Request 55: 194.97ms
Request 56: 196.30ms
Request 57: 193.27ms
Request 58: 194.15ms
Request 59: 267.06ms
Request 60: 197.03ms
Request 61: 197.72ms
Request 62: 193.77ms
Request 63: 192.98ms
Request 64: 205.65ms
Request 65: 194.46ms
Request 66: 193.42ms
Request 67: 193.46ms
Request 68: 197.18ms
```

7.



The Histogram (top chart)

Most of the requests (~125 out of ~150) completed in around 190-200ms. This is the baseline latency — mostly network round-trip time from my location to Oregon. The "long tail" on the right shows a few outliers that took 250-500ms. This is the classic distribution mentioned in the assignment.

The Scatter Plot (bottom chart)

Response times are fairly consistent throughout the 30 seconds, hovering around 200ms. The occasional spikes (like that ~490ms one around request 120) appear randomly — they're not clustered at the beginning or end, which suggests they're not caused by server warm-up or overload.

Answering against the parameters mentioned in the assignment:

1. **Distribution Shape:** Yes, it shows a long tail. Roughly 10-15% of requests fall into the "slow" category (above 210ms).
2. **Consistency:** Mostly consistent. The spikes appear randomly scattered, not in a pattern.
3. **Percentiles:** The median is probably around 195ms, and 95th percentile around 250-300ms. That's a moderate gap — some variability but not extreme.

4. **Infrastructure Impact:** A t2.micro is a shared-tenancy instance with burstable CPU. When other tenants on the same physical host are busy, or when the CPU credits fluctuate, occasional slower responses are obtained.
 5. **Scaling Implications:** With 100 concurrent users, we would likely see more contention — higher average latency and more frequent spikes. The t2.micro would become a bottleneck.
 6. **Network vs. Processing:** The ~190ms baseline is almost entirely network latency (the system → Oregon and back). The spikes are likely a mix of network jitter and occasional server-side delays.
- 8. Server-Side Request Logs** The Go server logs showing incoming requests from IP 155.33.133.38 (my machine). Server-side processing time is extremely fast (40-90 microseconds per request), confirming that the ~200ms response time measured by the client is almost entirely network latency, not server processing.

| |
|---|
| [GIN-debug] [WARNING] You trusted all proxies, this is NOT safe. we recommend you to set a value. |
| Please check https://github.com/gin-gonic/gin/blob/master/docs/doc.md#dont-trust-all-proxies for details. |
| [GIN-debug] Listening and serving HTTP on 0.0.0.0:8080 |
| [GIN] 2026/01/19 - 21:33:37 200 72.758µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:12 200 66.595µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:12 200 69.832µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:12 200 42.397µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:12 200 43.517µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:12 200 42.101µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:13 200 41.783µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:13 200 68.101µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:13 200 39.802µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:13 200 42.635µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:13 200 74.768µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:14 200 43.224µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:14 200 40.753µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:14 200 40.365µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:14 200 70.74µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:15 200 41.749µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:15 200 78.921µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:15 200 49.209µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:15 200 41.076µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:15 200 42.029µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:16 200 62.737µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:16 200 69.032µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:16 200 40.324µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:16 200 41.679µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:16 200 50.707µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:17 200 40.934µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:17 200 39.975µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:17 200 43.017µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:17 200 41.421µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:17 200 44.562µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:18 200 40.522µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:18 200 65.353µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:18 200 62.517µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:18 200 68.652µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:18 200 92.404µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:18 200 44.417µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:19 200 83.785µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:19 200 42.087µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:19 200 64.973µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:19 200 42.34µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:20 200 42.358µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:20 200 45.992µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:20 200 43.214µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:20 200 81.412µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:20 200 47.905µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:21 200 42.478µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:21 200 42.576µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:21 200 53.916µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:21 200 40.101µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:21 200 66.904µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:22 200 83.563µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:22 200 66.491µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:22 200 37.8µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:22 200 37.827µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:22 200 38.661µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:23 200 38.646µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:23 200 38.363µs 155.33.133.38 GET "/albums" |
| [GIN] 2026/01/19 - 21:44:23 200 39.138µs 155.33.133.38 GET "/albums" |

9. Load Test Statistics Final statistics from the load test: 148 total requests in 30 seconds, with an average response time of 203.85ms and median of 196.97ms. The 95th percentile (244.95ms) and 99th percentile (295.61ms) show moderate tail latency. The maximum response time was 492.28ms.

REQUEST 148: 198.10ms

Statistics:

Total requests: 148

Average response time: 203.85ms

Median response time: 196.97ms

95th percentile: 244.95ms

99th percentile: 295.61ms

Max response time: 492.28ms