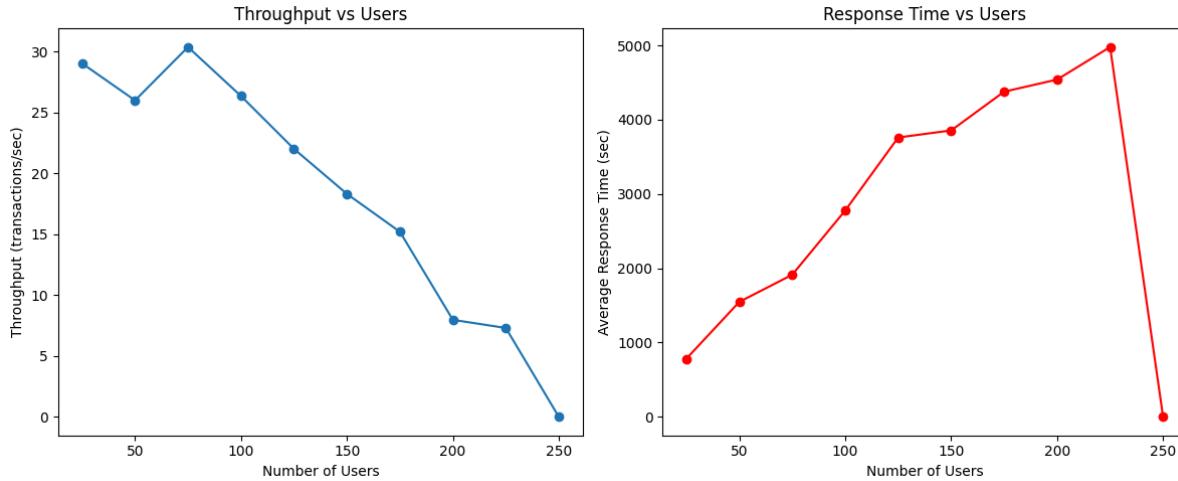


PaaS Auto Scaling

Baseline performance of PaaS



- From part D, show the inflection point on both of your curves and explain how you map this to the appropriate system resources (performance metrics) to use to trigger auto-scaling for PaaS.

Ans: The inflection point is when the number of concurrent users is 75 users the throughput starts to drop. At that time the CPU Utilization is around 35% so I set the Upper threshold at 30% and lower at 20%

- Describe the scaling policy you configured to scale up and scale down.

Ans: I configured Scaling cooldown to 60 second, Metric is CPU Utilization since our web app is matrix calculation which use CPU, Statistic to Maximum, Breach duration to 1 Min, Upper threshold 30%, and Lower threshold 20% this make it's very sensitive when our instance CPU Util reach the threshold 30% for 1 minute it's will trigger scaling operation.

Capacity			
Environment type		Min instances	Max instances
Load balanced		1	4
Fleet composition		On-demand base	On-demand above base
On-Demand Instances		0	0
Capacity rebalancing		Scaling cooldown	Processor type
Disabled		60	x86_64
Instance types		AMI ID	Availability Zones
t3.micro		ami-0736ae53e38c421d9	Any
Metric		Statistic	Unit
CPUUtilization		Maximum	Percent
Period		Breach duration	Upper threshold
1		1	30
Scale up increment		Lower threshold	Scale down increment
1		20	-1

3. How did you run siege (what options) to trigger scaling up and scaling down?

Ans: I ran `siege -c250 -t999H -b http://act-3-php-env.eba-tjajib2j.us-east-1.elasticbeanstalk.com/`

This with 250 concurrent users to allow CPU Utilization to reach above 30% and time for 999 hours or occur until i interrupt execution this is to make sure the CPU utilization will be above 30% for more than 1 minute.

4. Was the scaling up and scaling down behavior consistent with your scaling policy?

Ans: Yes, the scale up and down is consistent with my policy. Scale up when CPU Utilization reaches 30% and Scale down when it is below 20%.

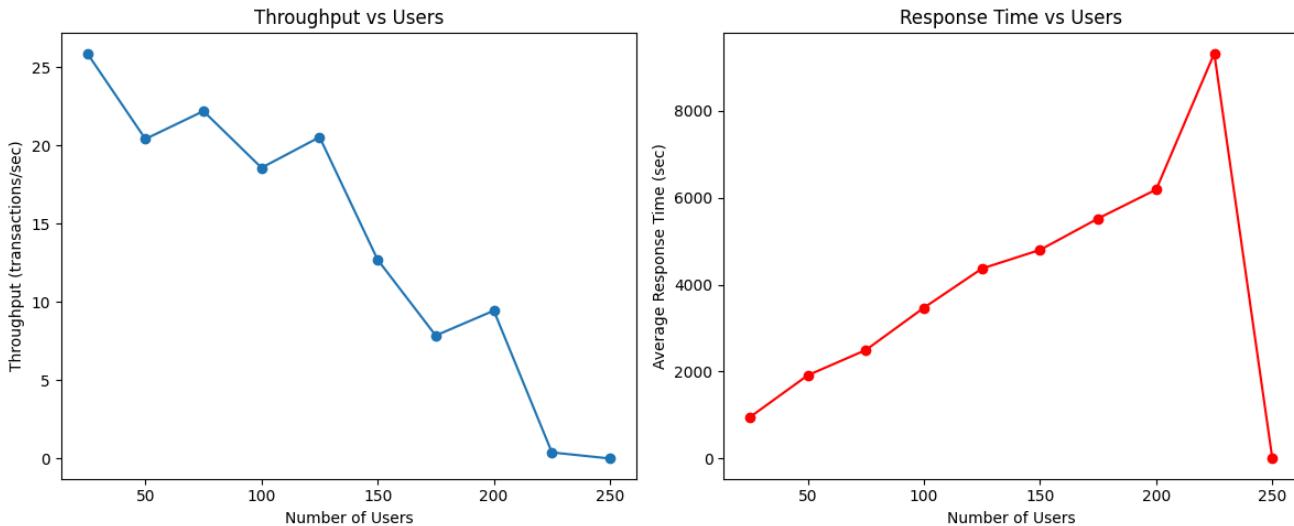
5. Include screenshots from Elastic Beanstalk showing the number of instances running and screenshots of resource monitoring data for your environment in your report to confirm that you successfully triggered auto-scaling according to your configured policies.

Ans: Here is the screen shot of the Event log which shows the ec2 instance added and removed.

Time	Type	Details
January 26, 2026 13:16:28 (UTC+7)	INFO	Removed Instance 'i-058c3aa310335983' from your environment. (Reason: Instance is in 'terminated' state)
January 26, 2026 13:14:29 (UTC+7)	INFO	Removed Instance 'i-0381e353ecc37bd5' from your environment. (Reason: Instance is in 'terminated' state)
January 26, 2026 13:13:29 (UTC+7)	INFO	Removed Instance 'i-092767233498998ad8f' from your environment. (Reason: Instance is in 'terminated' state)
January 26, 2026 13:00:30 (UTC+7)	INFO	Added EC2 Instance 'i-03af73f794001299' to Auto Scaling Group 'awseb-e-cb6pn2h2j-stack-AWSEBAutoScalingGroup-kWFrmIrcGpVW'.
January 26, 2026 13:00:30 (UTC+7)	INFO	Adding instance 'i-03af73f794001299' to your environment.
January 26, 2026 12:59:30 (UTC+7)	INFO	Adding instance 'i-09276723498996e0f' to your environment.
January 26, 2026 12:59:30 (UTC+7)	INFO	Added EC2 Instance 'i-09276723498996e0f' to Auto Scaling Group 'awseb-e-cb6pn2h2j-stack-AWSEBAutoScalingGroup-kWFrmIrcGpVW'.
January 26, 2026 12:56:30 (UTC+7)	INFO	Added EC2 Instance 'i-0391e53ee537b85' to Auto Scaling Group 'awseb-e-cb6pn2h2j-stack-AWSEBAutoScalingGroup-kWFrmIrcGpVW'.
January 26, 2026 12:56:30 (UTC+7)	INFO	Adding instance 'i-0391e53ee537b85' to your environment.
January 26, 2026 12:41:02 (UTC+7)	INFO	Environment update completed successfully.
January 26, 2026 12:41:01 (UTC+7)	INFO	Successfully deployed new configuration to environment.
January 26, 2026 12:40:23 (UTC+7)	INFO	Updating environment Act-3-php-env's configuration settings.
January 26, 2026 12:40:16 (UTC+7)	INFO	Environment update is starting.
January 26, 2026 12:07:41 (UTC+7)	INFO	Environment update completed successfully.
January 26, 2026 12:07:41 (UTC+7)	INFO	Successfully deployed new configuration to environment.
January 26, 2026 12:07:05 (UTC+7)	INFO	Updating environment Act-3-php-env's configuration settings.
January 26, 2026 12:06:55 (UTC+7)	INFO	Environment update is starting.

IaaS Auto Scaling

Baseline Performance of IaaS



1. Consider observed baseline performance in terms of response time and throughput as a function of offered load (# of client requests/second) when you deploy the same web app to PaaS vs. IaaS. Double-check to see if they are deployed on the same actual instance sizes and check the AZ.

1.1. Discuss the differences in baseline performance in terms of response time and throughput as a function of offered load (# of client requests/second), and conclude which provides better performance?

Ans: From the baseline performance results, PaaS (Elastic Beanstalk) performed better than IaaS. The average response time for PaaS was significantly lower than the manual IaaS setup under the same load conditions.

1.2. Are there any reasons that you think would sensibly explain the performance differences?

Ans: I think it's because PaaS(Elastic Beanstalk) is using Nginx as a reverse proxy and has OPCache which results in Elastic Beanstalk processes the request and returns the HTML faster than untuned Apache server(IaaS).

2. For IaaS, what scaling policy did you use to scale up and scale down?

Ans:

- Scale Up: Triggered when Maximum CPU Utilization is above 70% for 30 seconds.
 - Scale Down: Triggered when Maximum CPU Utilization is below 30% for 30 seconds.
-

3. How did you run siege (what options) to trigger scaling up and scaling down?

Ans: For scale up I use `siege -c 250 -t 30M -b http://phiaasgroup-1-407212990.us-east-1.elb.amazonaws.com/`.

This trigger scale up because it increase CPUUtilization to be above 70% for more than 30 seconds and for scale down I use `siege -c 20 -t 3M -v http://phiaasgroup-1-407212990.us-east-1.elb.amazonaws.com/`.

This trigger scale down because it didn't use that much CPUUtilization(below 30%) for 30 seconds

4. Was the scaling up and scaling down behavior consistent with your scaling policy?

Ans: Yes, the behavior was consistent. The activity logs show new instances launching shortly after the CPU alarm was triggered, and instances being terminated when the load decreased.

5. Include screenshots from EC2 monitoring data in the report to make sure I can see that you did trigger auto-scaling according to your configured policies.

Instances (3/6) info														Last updated 1 minute ago	Connect	Instance state	Actions	Launch instances
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 IP	Elastic IP	IPv6 IPs	Monitoring	Security group name	Key name					
i-0da9165180ea98bf	i-0da9165180ea98bf	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-	-	-	disabled	-	cu_cloud					
i-0fbc52445e11303a5	i-0fbc52445e11303a5	Running	t2.micro	3/3 checks pass	View alarms +	us-east-1a	ec2-44-198-158-225.co...	44.198.158.225	-	-	disabled	launch-wizard-1,ec2-rds-1	cu_cloud					
act-5-siege	i-015f5fbfa1029e72f	Running	t2.micro	3/3 checks pass	View alarms +	us-east-1a	ec2-44-202-196-66.co...	44.202.196.66	-	-	disabled	launch-wizard-1	cu_cloud					
Act-3-php-env	i-0290a1545affa1f0a	Running	t2.micro	3/3 checks pass	View alarms +	us-east-1a	-	-	-	-	disabled	launch-wizard-1,awselb...	cu_cloud					
act-5-php	i-03ed48bcdc409172c	Terminated	t2.micro	-	View alarms +	us-east-1a	-	-	-	-	disabled	-	cu_cloud					
	i-061ad1b6696d20b35	Running	t2.micro	3/3 checks pass	View alarms +	us-east-1a	ec2-44-197-216-55.co...	44.197.216.55	-	-	disabled	launch-wizard-1,ec2-rds-1	cu_cloud					

(3 selected instances are managed by AutoScalingGroup named "phiaasgroup")

Auto Scaling groups (1/2) info														Last updated 46 minutes ago	Launch configurations	Launch templates	Actions	Create Auto Scaling group
Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time										
phiaasgroup	phiaasappaunch Version Default	1	-	1	1	4	user1-az1 (us-east-1)	Mon Jan 26 2026 15:46:52 GMT+0700 (Indochina Time)										
	aws-e-cilbgp2h2j1-stack-AWSEBAutoScalingGroup-LWTHmtrzrePw	AWSLaunchTemplate_5tHQnctT0V1	1	-	1	4	user1-az1 (us-east-1)	Sun Jan 26 2026 22:16:26 GMT+0700 (Indochina Time)										
Auto Scaling group: phiaasgroup																		
Status	Description	Cause		Start time		End time												
Successful	Terminating EC2 instance: i-05ed48bcdc409172c	At 2026-01-26T10:55:54Z a monitor alarm phiaasapp scale-down alarm in state ALARM triggered policy scale-down changing the desired capacity from 2 to 1. At 2026-01-26T10:41:03Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2026-01-26T10:41:03Z instance i-03ed48bcdc409172c was selected for termination.		2026 January 26, 05:41:03 PM +07:00		2026 January 26, 05:46:46 PM +07:00												
Successful	Terminating EC2 instance: i-0da9165180ea98bf	At 2026-01-26T10:55:54Z a monitor alarm phiaasapp scale-down alarm in state ALARM triggered policy scale-down changing the desired capacity from 3 to 2. At 2026-01-26T10:55:59Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2026-01-26T10:36:00Z instance i-0da9165180ea98bf was selected for termination.		2026 January 26, 05:36:00 PM +07:00		2026 January 26, 05:42:23 PM +07:00												
Successful	Launching a new EC2 instance: i-09e52445613305a5	At 2026-01-26T10:50:58Z a monitor alarm phiaasapp scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 2 to 3. At 2026-01-26T10:31:06Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.		2026 January 26, 05:31:08 PM +07:00		2026 January 26, 05:31:13 PM +07:00												
Successful	Launching a new EC2 instance: i-05d9165180ea98bf	At 2026-01-26T10:55:54Z a monitor alarm phiaasapp scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 1 to 2. At 2026-01-26T10:26:02Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.		2026 January 26, 05:26:04 PM +07:00		2026 January 26, 05:26:11 PM +07:00												
Successful	Updating load balancers/target groups: Success! Status: Running. Address: amazeeelastisloadbalancergroup-east-1:1024-1772483-target-i-0edc1db77248492 (Target Group).			2026 January 26, 05:46:46 PM +07:00		2026 January 26, 05:46:46 PM +07:00												
Successful	Launching a new EC2 instance: i-05ed48bcdc409172c	At 2026-01-26T08:46:52Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2026-01-26T08:46:52Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.		2026 January 26, 05:46:37 PM +07:00		2026 January 26, 05:46:43 PM +07:00												

(Activity logs of Auto Scaling group "phiaasgroup")

6. Consider what Amazon did when you terminated your EC2 instance(s) in part D (fault tolerance).

(terminated an instance)

6.1. Is Amazon able to launch new instances to replace the one(s) you killed?

Ans: Yes. The Auto Scaling Group detected the unhealthy (terminated) instance and automatically launched a new replacement instance to maintain the desired capacity.

Status	Description	Cause	Start time	End time
Successful	Launching a new EC2 instance: i-008d6b9f57b957235	At 2026-01-26T11:16:18Z a monitor alarm phlaasgroup scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 2 to 3. At 2026-01-26T11:16:26Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2026 January 26, 06:16:26 PM +07:00	2026 January 26, 06:16:33 PM +07:00
Successful	Launching a new EC2 instance: i-008d6b9f57b957235	At 2026-01-26T11:16:25Z an instance was launched in response to an unhealthy instance needing to be replaced.	2026 January 26, 06:16:27 PM +07:00	2026 January 26, 06:16:33 PM +07:00
Connection draining in progress	Terminating EC2 instance: i-008d6b9f57b957235 - Waiting for ELB Connection Draining.	At 2026-01-26T11:16:25Z an instance was taken out of service in response to an EC2 health check indicating that it has been terminated or stopped.	2026 January 26, 06:16:25 PM +07:00	
Successful	Launching a new EC2 instance: i-008d6b9f57b957235	At 2026-01-26T11:16:18Z a monitor alarm phlaasgroup scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 1 to 2. At 2026-01-26T11:16:32Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2026 January 26, 06:11:33 PM +07:00	2026 January 26, 06:11:39 PM +07:00
Successful	Terminating EC2 instance: i-03ed48b0cd409172c	At 2026-01-26T10:04:54Z a monitor alarm phlaasgroup scale down alarm in state ALARM triggered policy scale-down changing the desired capacity from 2 to 1. At 2026-01-26T10:04:53Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 2 to 1. At 2026-01-26T10:41:03Z instance i-03ed48b0cd409172c was selected for termination.	2026 January 26, 05:41:03 PM +07:00	2026 January 26, 05:46:46 PM +07:00
Successful	Terminating EC2 instance: i-0ed9f165180e998f	At 2026-01-26T10:55:42Z a monitor alarm phlaasgroup scale down alarm in state ALARM triggered policy scale-down changing the desired capacity from 3 to 2. At 2026-01-26T10:55:42Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 3 to 2. At 2026-01-26T10:56:02Z instance i-0ed9f165180e998f was selected for termination.	2026 January 26, 05:36:00 PM +07:00	2026 January 26, 05:42:23 PM +07:00
Successful	Launching a new EC2 instance: i-0ed9f165180e998f	At 2026-01-26T10:55:42Z a monitor alarm phlaasgroup scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 2 to 3. At 2026-01-26T10:55:42Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 2 to 3.	2026 January 26, 05:31:08 PM +07:00	2026 January 26, 05:31:13 PM +07:00
Successful	Launching a new EC2 instance: i-0ed9f165180e998f	At 2026-01-26T10:25:58Z a monitor alarm phlaasgroup scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 1 to 2. At 2026-01-26T10:26:02Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2026 January 26, 05:26:04 PM +07:00	2026 January 26, 05:26:11 PM +07:00
Successful	Upgrading load balancer target groups: Successful, Status Reason: Added: amazwselasticloadbalancergroup:1/01234554485/target group/phlaasgroup-1/0ed9f165180e998f [Target Group]	At 2026-01-26T03:46:46Z a monitor alarm phlaasgroup scale up alarm in state ALARM triggered policy scale-up changing the desired capacity from 1 to 2. At 2026-01-26T03:46:46Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 1 to 2.	2026 January 26, 03:46:46 PM +07:00	2026 January 26, 03:46:46 PM +07:00

(Activity history that Auto Scaling group launching new instances 2 upper rows)

The screenshot shows the AWS EC2 Instances page. A search bar at the top has 'Search' and '[Alt+I]' placeholder text. Below it is a navigation bar with 'EC2' and 'Instances'. The left sidebar contains sections for Dashboard, Events, Instances, 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, Network & Security, Security Groups, Placement Groups, Key Pairs, Network Interfaces, Load Balancing, Target Groups, Trust Stores, and Auto Scaling. The 'Auto Scaling' section is expanded, showing 'Auto Scaling Groups' and 'Settings'. The main content area displays a table titled 'Instances (1/9) info' with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4 IP, Elastic IP, IPv6 IPs, Monitoring, Security group name, and Key name. There are 10 rows of data, with the last row (i-0d8205bd677c8efbb) being the selected one. At the bottom of the table is a horizontal scrollbar.

(The lowest and selected row is the latest instances that got launched)

6.2. When the new instances are launched, are your web clients (browser or siege) able to access the web application like normal? Provide evidence of what your web clients see.

Ans: Yes. Once the new instance passed its health checks, the Load Balancer resumed sending traffic to it. My Siege client and browser could access the web application normally without manual intervention.

The screenshot shows the AWS CloudFront distribution configuration for 'My PHP App'. The 'General' tab is active, showing settings for 'Origin' (Amazon RDS), 'Default Cache Behavior' (Cache Only), and 'Compress' (Yes). The 'Behaviors' tab lists two behaviors: 'All requests' (Cache Only) and 'GET /' (Cache Only). The 'Logs' tab shows log prefix '/var/log/cloudfront' and log file format. The 'Metrics' tab shows metrics for CloudWatch Metrics.

What's Next?

- AWS Elastic Beanstalk overview
- Deploying AWS Elastic Beanstalk Applications in PHP Using Eb and Git
- Using Amazon RDS with PHP
- Customizing the Software on EC2 Instances
- Customizing Environment Resources

AWS SDK for PHP

- AWS SDK for PHP home
- PHP developer center
- AWS SDK for PHP on GitHub

```
ec2-user@ip-172-31-15-173:~$ curl -v http://44.192.37.92/
* Rebuilt URL to: http://44.192.37.92/
*   Trying 44.192.37.92...
* TCP_NODELAY set
* Connected to 44.192.37.92 (44.192.37.92) port 80 (#0)
* HTTP request method GET
* Host: 44.192.37.92
* User-Agent: curl/7.54.0
* 
< HTTP/1.1 200 0.09 secs: 1438 bytes ==> GET /
< HTTP/1.1 200 0.02 secs: 193 bytes ==> GET /css?family=Lobster+Two
< HTTP/1.1 200 0.10 secs: 1438 bytes ==> GET /
< HTTP/1.1 200 0.19 secs: 1438 bytes ==> GET /
< HTTP/1.1 200 0.10 secs: 3027 bytes ==> GET /
< HTTP/1.1 200 0.00 secs: 3498 bytes ==> GET /styles.css
< HTTP/1.1 200 0.02 secs: 3027 bytes ==> GET /
< HTTP/1.1 200 0.02 secs: 193 bytes ==> GET /css?family=Lobster+Two
< HTTP/1.1 200 0.01 secs: 3498 bytes ==> GET /styles.css
< HTTP/1.1 200 0.10 secs: 3027 bytes ==> GET /
< HTTP/1.1 200 0.01 secs: 3498 bytes ==> GET /styles.css
< HTTP/1.1 200 0.03 secs: 193 bytes ==> GET /css?family=Lobster+Two
< HTTP/1.1 200 0.02 secs: 193 bytes ==> GET /css?family=Lobster+Two
< HTTP/1.1 200 0.12 secs: 3027 bytes ==> GET /
< HTTP/1.1 200 0.00 secs: 3498 bytes ==> GET /styles.css
< HTTP/1.1 200 0.02 secs: 193 bytes ==> GET /css?family=Lobster+Two
< HTTP/1.1 200 0.11 secs: 1438 bytes ==> GET /
< HTTP/1.1 200 0.10 secs: 3027 bytes ==> GET /
```

7. Discuss the differences between configuring auto-scaling in IaaS vs. auto-scaling PaaS. What are the strengths of each configuration approach?

Ans: Both can achieve the same configuration but there are some differences.

- **PaaS (Elastic Beanstalk):** Configuration is much simpler. All settings (scaling triggers, load balancing, health checks) are centralized on a single "Configuration" dashboard.
 - **IaaS (Manual):** Requires more manual steps. I had to separately configure the Launch Template, Target Groups, Load Balancer, and Auto Scaling Group policies.
-

8. Discuss the differences between IaaS vs. PaaS. What are the strengths of each approach?

Ans: I think both have trade offs.

- **IaaS Strengths:** Offers higher flexibility and control. We can customize the OS, install specific kernel modules, and tune the network stack exactly how we need it.
 - **PaaS Strengths:** Offers higher convenience and speed. It handles deployment, patching, and scaling logic automatically, allowing developers to focus on code rather than infrastructure.
-