Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT490-450-M2024/it490-module-2-individual-research-and-example/grade/lcl

IT490-450-M2024 - [IT490] Module 2 Individual Research and Example

Submissions:

Submission Selection

1 Submission [active] 5/31/2024 9:27:11 AM

Instructions

COLLAPSE

Overview Video: https://youtu.be/tzk4ewLSaDI

1. Create a new branch following the desired branch name and replace ucid with your ucid
2. Investigate a few vendors (i.e., Google, Amazon, Microsoft, Oracle) and explore their Virtual Machine instance offerings (like EC2 instances)
    1. Requirements to look for (note: Heroku won't be an option)
        1. Affordability (or Free tier rules) and offered free credits for a specific duration
        2. Ubuntu is an option
        3. You have root access
        4. You can control the firewall rules
        5. Ensure you can have multiple VMs managed under the same "free credit" quota (usually you can)
3. Create 1 VM under your chosen cloud provider
4. Get the example code working and capture evidence of it working
5. Fill in the below deliverables
6. Export the PDF and add it to this branch
7. Add/commit/push your changes
8. Create a pull request for this branch and merge the code to the primary branch
9. Upload the PDF to Canvas
10. You may want to turn off your server so you don't waste any quota once you're done

Branch name: M2-Example-ucid

Tasks: 7 Points: 10.00

Research (3.5 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Mention the Vendors you explored and some pros/cons for each that affected your decision

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	At least two vendor options compared
#2	1	Clearly shows pros/cons (3 of each)

Response:

The two vendors I looked at were Amazon Web Services and Micorsoft Azure. Microsoft Azure Pros: Azure Virtual Machines free for 12 months, have small experience with Azure, option of Windows or Linux Microsoft Azure Cons: root user is typically disabled by default, need to switch to root user, firewall is not free Amazon Web Services Pros: have root access, uses Ubuntu, easy to use Amazon Web Services Cons: data transfer: 1.0 GB are always free per month, Elastic Compute Cloud: 750.0 Hrs for free for 12 months , need to pay attention to these so I don't have to pay for each month

COLLAPSE

Task #2 - Points: 1

Text: Which vendor option did you go with and why?

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Vendor mentioned
#2	1	Clear reason for choice

Response:

I chose Amazon Web Services as my vendor. This is because it was user friendly and easy to figure out how to use. I also didn't have to download anything to use or access it. Also, I can connect using EC2 instance connect on the web and use Ubuntu instead of connecting to my laptop's command prompt or Gitbash.

COLLAPSE

Task #3 - Points: 1

Text: How do you plan to manage your usage quota? (Also mention the quota)

Checklist

*The checkboxes are for your own tracking

#	Points	Details

#1	1	Quota limit is mentioned clearly (what are your restrictions)
#2	1	Logical handling of quota mentioned (i.e. does the vendor provide an automated way, will it be manual, etc)

Response:

The quota limit for Amazon Web Services are 50,000 AMIs, 5 EC2-VPC Elastic IPs, 10,000 Launch template versions and 5,000 Launch templates. On AWS there is a way to check your service quotas. I plan to use that as well as manually calcualte it.

Example Evidence (3.5 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshot of the sample request being published and receiving a reply

Details:
This is the publisher perspective

#1) Valid Request sent (you may need to modify the code to show this)

```
ubuntu@ip-172-31-9-171:~/IT490$ php RabbitMQClientSample.php
sending data: Array
(
    [message] => message
    [type] => echo
)
sending message
sending message
object(stdClass)#27 (2) {
  ["return_code"]=>
  string(1) "0"
  ["message"]=>
  string(13) "Echo: message"
}
client received response:
stdClass Object
(
    [return_code] => 0
    [message] => Echo: message
)
stdClass Object
(
    [return_code] => 0
    [message] => Echo: message
)
RabbitMQClientSample.php END
ubuntu@ip-172-31-9-171:~/IT490$
```

Caption (required) ✓

Describe/highlight what's being shown

showing valid request sent

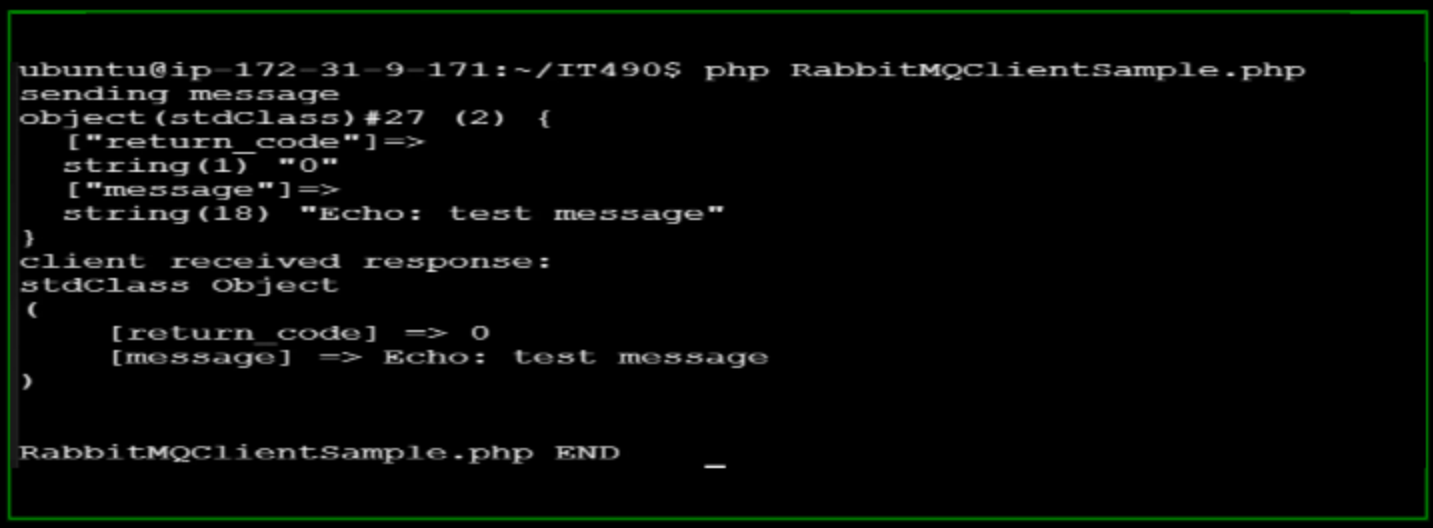
Explanation (required) ✓

Explain what was edited to get this to show

EDIT RESPONSE

I edited the echo message for the data being sent and the message being sent.

#2) Valid Response Received



Caption (required) ✓

Describe/highlight what's being shown

showing response received

^COLLAPSE ^

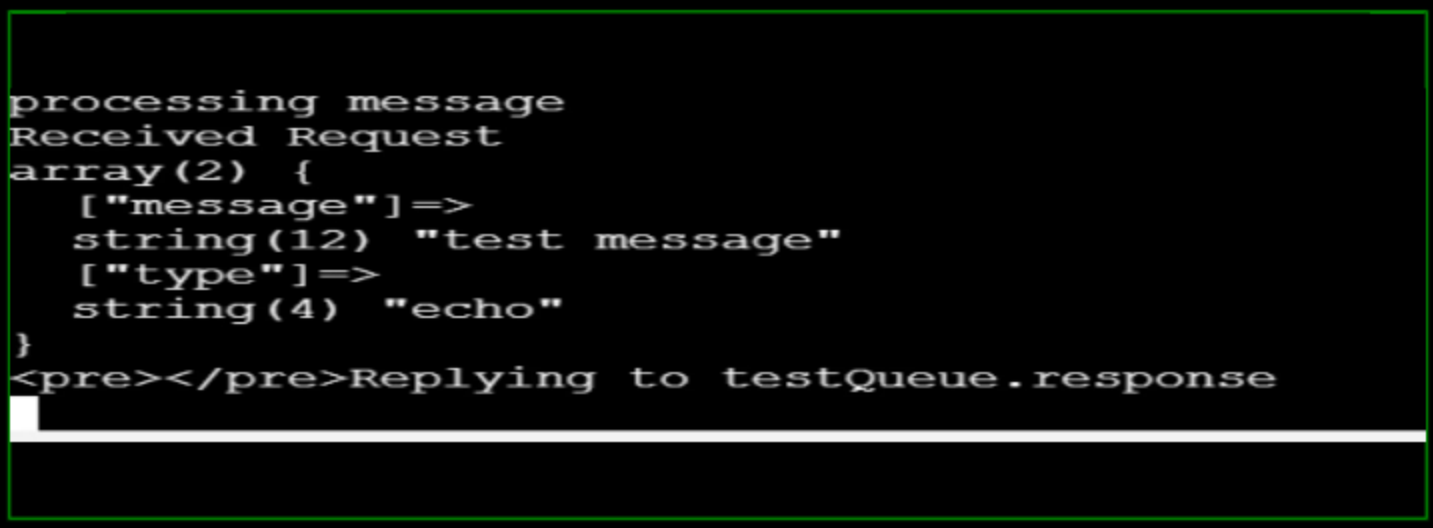
Task #2 - Points: 1

Text: Screenshot of the consumer receiving the request and replying back

Details:

This is the consumer perspective

#1) Valid Request Received



Caption (required) ✓

Describe/highlight what's being shown

showing valid request received

#2) Valid Response sent (you may need to modify the code to show this on the terminal)

```
ubuntu@ip-172-31-9-171:~/IT490$ php RabbitMQServerSample.php
Rabbit MQ Server Start
processing request
Consuming queue
processing message
Received Request Array
(
    [message] => message
    [type] => echo
)
 <pre></pre>
Sending response message: Array
(
    [return_code] => 0
    [message] => Echo: message
)

Replying to testQueue.response
```

Caption (required) ✓

Describe/highlight what's being shown

showing valid response sent

Explanation (required) ✓

Explain what was edited to get this to show

EDIT RESPONSE

I edited the echo message to print the response being sent and return response.

Discussion (2 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: What issues did you face and how did you resolve them?

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	At least one issue clearly mentioned
#2	1	Clear solution mentioned for issue(s)

Response:

An issue I faced was at first the client and server were not sending and recieving the message. To resolve it I used the

SSH link instead of the HTTP link and created a SSH key in GitHub. After completing that, the client and server were working. Another issue I faced was figuring out how to modify the code to show a valid request sent and valid repsonse sent. To resolve it I tried to visualize the client and server tallking to each other and who would send and recieve the message.

Misc (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Pull Request Link for this assignment (should end in /pull/#)

Details:

Valid pull request that ends in /pull/#

URL #1

Missing URL

End of Assignment