Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT490-451-M2024/it490-module-2-individual-research-and-example/grade/ms268

IT490-451-M2024 - [IT490] Module 2 Individual Research and Example

Submissions:

Submission Selection

1 Submission [active] 5/29/2024 7:33:06 PM

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:

I researched the 3 major ones (AWS), GCP, and Microsoft Azure, and explored all their features, pricing, performance, and support. Below are some of the pros and cons.

AWS Pros Mature Ecosystem - has a really good extensive ecosystem with some third-part integrations some offered even at the free level Scalability - It allowed us to create an organization without much hassle and has some other features Auto Scaling and Elastic load balance, not sure if both are available for free users though

Cons - There seems to be a learning curve because the platform is really big Compared to the rest it is costly so we need to make sure that we are not being charged

GCP - Pros - Had better performance and strong data analytics capabilities Good networking opportunities and better speed as a free account

Cons - Services and instances type were lower compared to AWS however the main EC2 was there but wasn't sure what we would need in the future Ecosystem - a smaller ecosystem and community compared to AWS, was able to find a lot more documentation on AWS than on GCP online

Azure Pros - Integration with other Microsoft products, meeting on Teams and SQL server would have been amazing We have free azure accounts from the university even though we have to request them.

Cons- Similar to AWS the price here was expensive too if we got charged Learning Curve is there and also read some performance issues from users on the web

● ^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:

What I choose and why - Decided to go with AWS just because it seemed a lot easier to make it work with the team, some of the memebrs had some experience and were willing to teach the rest of us. Additonally the professor also had a video explaining how to make it work on AWS. There was also some good study material that we got access to from AWS Educate. Finally it just seemed like a no brainer we were getting about 750 hours for free.

●
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:

I plan to manage the usage quote for AWS services by making sure I only use it when I need to and making sure I stop the instance when done working, additionally, the AWS free tier offers up to 750 hours of public IPv4 address usage per month for the next 12 months, however, I am still going to make sure I monitor and manage it effectively. AWS also provides some of the rules to help allow me to keep track however a lot of them are blocked because I am on the free account. There most of the dashboard settings are locked however I have access to all the essentials like starting a server, and usage limits. They also send an email when the usage is about 85%. There are some manual adjustments too however I am still learning a lot about what AWS offers. We also got access to their education website which has a lot of studying material about AWS which was also a winning factor.

●
Example Evidence (3.5 pts.)
COLLAPSE

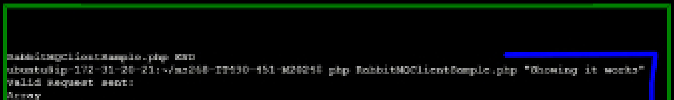
●
COLLAPSE

Task #1 - Points: 1

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

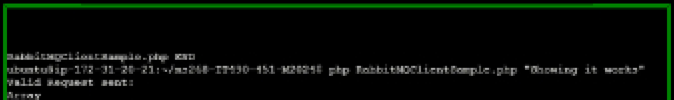
Details:
This is the publisher perspective

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

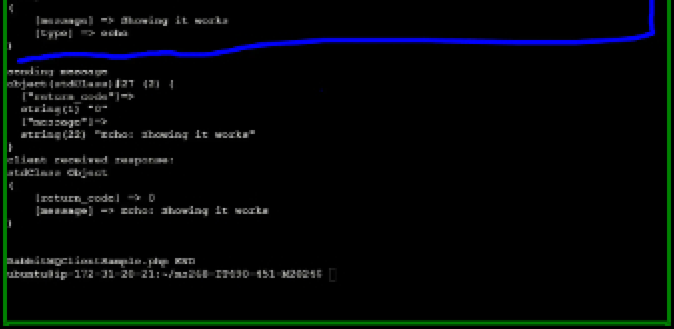


```
RabbitMQClientSample.php END
ubuntu@ip-172-31-20-21:~/ms268-IT490-451-M2024$ php RabbitMQClientSample.php "Showing it works"
Valid Request sent:
Array
```

#2) Valid Response Received

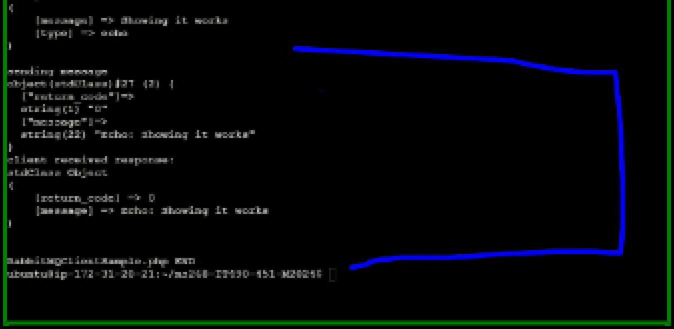


```
RabbitMQClientSample.php END
ubuntu@ip-172-31-20-21:~/ms268-IT490-451-M2024$ php RabbitMQClientSample.php "Showing it works"
Valid Request sent:
Array
```



```
(
    [message] => Showing it works
    [type] => echo
)
sending message
object(stdClass)#27 (2) {
  ["return_code"]=>
  string(1) "0"
  ["message"]=>
  string(22) "Echo: Showing it works"
}
client received response:
stdClass Object
(
    [return_code] => 0
    [message] => Echo: Showing it works
)


RabbitMQClientSample.php END
ubuntu@ip-172-31-20-21:~/as268-IT490-451-M2024S
```



```
(
    [message] => Showing it works
    [type] => echo
)
sending message
object(stdClass)#27 (2) {
  ["return_code"]=>
  string(1) "0"
  ["message"]=>
  string(22) "Echo: Showing it works"
}
client received response:
stdClass Object
(
    [return_code] => 0
    [message] => Echo: Showing it works
)


RabbitMQClientSample.php END
ubuntu@ip-172-31-20-21:~/as268-IT490-451-M2024S
```

Caption (required) ✓

Describe/highlight what's being shown

Showing Request sent

Caption (required) ✓

Describe/highlight what's being shown

showing request received as an object

Explanation (required) ✓

Explain what was edited to get this to show

PREVIEW RESPONSE

To include the message "Valid Request sent" in the client-side code, I added an echo statement just before the request is sent along with the details of the message. On the server, I then enhanced the debugging output by printing the received request and added some error messages if the message type was unsupported. These modifications improve the visibility and debugging capabilities of both client and server, aiding in tracking the flow of requests and responses within the system.

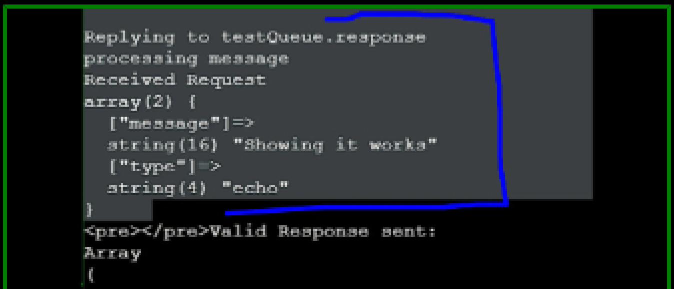
^COLLAPSE^

Task #2 - Points: 1

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

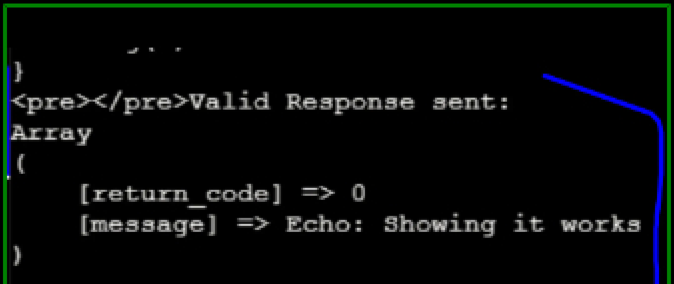
Details:
This is the consumer perspective

#1) Valid Request Received

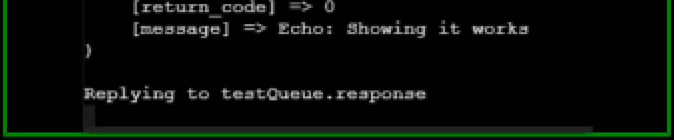


```
Replying to testQueue.response
processing message
Received Request
array(2) {
  ["message"]=>
  string(16) "Showing it works"
  ["type"]=>
  string(4) "echo"
}
<pre></pre>Valid Response sent:
Array
(
```

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

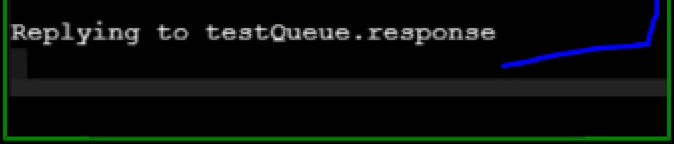


```
}
<pre></pre>Valid Response sent:
Array
(
    [return_code] => 0
    [message] => Echo: Showing it works
)
```



```
        [return_code] => 0
        [message] => Echo: Showing it works
)

Replying to testQueue.response
```



```
Replying to testQueue.response
```

Caption (required) ✓

Describe/highlight what's being shown

showing the same request having been received on the server side

Caption (required) ✓

Describe/highlight what's being shown

showing valid response sent

Explanation (required) ✓

Explain what was edited to get this to show

PREVIEW RESPONSE

With this one, I added an echo statement in the server-side code before returning the response. The line prints a message to the terminal indicating that a valid response is being sent, along with the other details using PHP_EOL, followed by a print statement similar to the one in our code already. Also included is a print error message if the type is unsupported.

● 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:

I faced a few issues while working on the RabbitMQ client-server setup. The first issue was the queue locking issue where the error "Resource_locked - cannot obtain exclusive access to locked queue 'testQueue.response' showed up and didn't let me continue. To fix that I just restarted my RabbitMQ which the professor later told me was a temp fix so I will try research on that for a more permanent fix. Secondly, i had an issue with the permission error while pushing my changes to GitHub which i resolved by configuring my SSH key correctly and ensuring the remote repo URL was set properly

● 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
https://github.com/Sameed-S/ms268-IT490-451-M2024/pull/4

End of Assignment