

COSC349 Assignment 1: Effecting the portable building and deployment of software applications using virtualisation.

Project Issues:

The current project does not function as desired. Currently the website server VM is having issues accessing any data from the database server regardless of it previously working in the past. The multi-selector, radio buttons and convert button present within the website server code, also do not show up on the website, regardless of previously working. This means that the user cannot currently see the contents of the database VM and cannot choose a country/city or conversion type. The lack of the “convert” button on the website server means that the user is unable to navigate to the conversion server site and thus cannot see the conversion results of the chosen country/city and conversion type.

I as a developer struggled with coding in php and found that very small errors, in my opinion, would cause breakdown of the whole project. Also towards the end of the project deadline, after rebuilding the project following a day of developing, I found that suddenly the database server would not connect to the website server, thus the database information could not be displayed on the website server. Due to this error I had little time to determine if the conversion VM code functioned appropriately and wrote the majority of the code while the conversion server had no user input to test with, due to the aforementioned error. I am still unsure of what causes this issue and hope future developers of this project will be able to find the cause of this bug, as it currently causes the breakdown of the whole project.

Figures comparing the previous working website server and the current website server functionality are present within the “Application Use” section of this report.

New Developer Explanation:

With a simple repository layout, and comments present in all program files, providing the developer is relatively familiar with the ideas of this project, it would be simple for them to understand the current state of the project and how the project currently functions.

Initially any new developers should attempt to fix the aforementioned bugs or rebuild the VM structure from scratch to ensure proper functionality of the VM interactions. Regardless of the mentioned errors, many developments can be made within this project, especially in terms of VM interaction.

I believe it would be beneficial for new developers to ensure that the conversion VM directly accesses the database server, even if it is just accessing the country/city names, so that the current country/city and conversion method hard coding is not required. Further accessing of the database server from the website server would also reduce hardcoding, especially within the multi-selection implementation.

The adding of the currency conversion rates and actual time zone differences to the database server, would allow the conversion VM to perform actual calculations and prevent hard coding from occurring. This could only occur if the conversion VM could directly interact with information found in the SQL table present on the database server.

Developers could also change the output of the conversion VM, by providing it with real-time currency conversions and current times of the country/city options, gained from another online server for example. This would allow output to be more informative to the users, and would make the provided information more accurate, especially when considering the currency conversion.

After each of these programming changes are made, git commits and pushes need to be made to ensure these changes are present in the project repository. If refreshing the website does not cause the changes to be shown in the associated servers, often the vagrant VM's need to be reloaded in order for these updates to be present. The **"vagrant destroy"** command line command can be used to completely shut off all of the VM's, the **"vagrant up --provider virtualbox"** command can then again be used to rebuild the project to completely ensure all changes are now going to be shown in their associated servers.

I also recommend future developers implement complete separation of the VM's to uphold security and reduce the effect of security breaches. The addition of extra hardware machines containing duplicates of the three interconnected VM's, could also increase the robustness of this project, as if hardware failures occur the websites and project in general would be able to continue to be run with little downtime.

Project Running:

To run the aforementioned project, navigate to this github repository:
https://github.com/kezia-petch/COSC349_Assign_1.

Clone the COSC349_Assign_1 repository, using HTTPS, found at the link above. Users will then need to clone the repository into their home directory, in order to actually access the project. This cloning process takes approximately 6 - 8 minutes and can be completed in terminal, using the command;

git clone [The HTTPS link provided when the user cloned the repository]

Then navigate to the COSC_Assign_1 folder that the previous git clone command would have created in your home directory. Then run the command, in terminal;

vagrant up --provider virtualbox

The user should then navigate to the webpage; <https://localhost:8090> to view the project in action.

This manual interaction is required to ensure that the vagrant project can function properly on the computer, and is very minimal.

Download Volumes:

The typical download size of three separate VM's is approximately 60MB. As this project is also using the Ubuntu xenial64 box, thus its approximate size of 270MB also needs to be taken into account when analysing the project's download volume.

The download volumes for both the first build and for subsequent builds will remain approximately the same, unless another VM is added into the project or an Ubuntu xenial64 box updated version is released and added to the project. Correct use of the vagrant destroy function should also be used after each session.

Application Use: Explained as through the project functions as desired

Currency and Timezone Converter

Showing a list of the possible Country and associated City conversions:

Country/City	Currency	Timezone	Offset from UTC
Australia/Sydney	Australian Dollar(AUD)	AEST	UTC +10 hours
Brazil/Brasilia	Real(BRL)	BRT	UTC -3 hours
Cabo Verde/Praia	Cabo Verdean Escudo(CVE)	CVT	UTC -2 hours
Canada/Whitehorse	Canadian Dollar(CAD)	PDT	UTC -9 hours
Chile/Easter Island	Chilean Peso(CLP)	EAST	UTC -7 hours
Cook Islands/Rarotonga	NZ Dollar(NZD)	CKT	UTC -11 hours
India/Kolkata	Indian Rupee(INR)	IST	UTC +6 hours
Kiribati/Kiritimati	Australian Dollar(AUD)	LINT	UTC +14 Hours
Maldives/Male	Rufiyaa(MVR)	MVT	UTC +5 hours
Malta/Valletta	Euro(EUR)	CEST	UTC +1 hour
NZ/Wellington	NZ Dollar(NSD)	NZST	UTC +12 hours
Oman/Muscat	Omani Rial(OMR)	GST	UTC +4 hours
Peru/Lima	Sol(PEN)	PET	UTC -5 hours
Samoa/Apia	Tala(WST)	WST	UTC +13 hours
Sierra Leone/Freetown	Leone(SLL)	GMT	UTC -1 hour
Singapore/Singapore	Singapore Dollar(SGD)	SGT	UTC +7 hours
South Africa/Johannesburg	Rand(ZAR)	SAST	UTC +2 hours
South Korea/Seoul	South Korean Won(KRW)	KST	UTC +9 hours
Taipei/Taiwan	New Taiwan Dollar(TWD)	CST	UTC +8 hours
United Kingdom/Bermuda	Bermudian Dollar(BMD)	ADT	UTC -4 hours
United Kingdom/Isle Of Man	Manx Pound(IMP)	BST	No Difference
United States/Fairbanks	United States Dollar(USD)	AKDT	UTC -10 hours
United States/Houston	United States Dollar(USD)	CDT	UTC -6 hours
United States/Los Angeles	United States Dollar(USD)	PDT	UTC -8 hours
Vanuatu/Port Vila	Vatu(VUV)	VUT	UTC +11 hours
Yemen/Sana	Yemeni Rial(YER)	AST	UTC +3 hours

Select What Country/City and Conversion Type You Want:

Country/City:

Australia/Sydney
Brazil/Brasilia
Cabo Verde/Praia
Canada/Whitehorse

Conversion Type:
☐ Currency Conversion
☐ Timezone Conversion

Currency and Timezone Converter

Showing the possible Country and associated City conversions:

Country/City	Currency	Timezone	Offset from UTC
--------------	----------	----------	-----------------

Figure 2: Current broken project image.

Figure 1: Previous working project image.

The current broken project implementation and previous functioning implementation are both represented in the figures above. If the project were to function as desired, the user would be displayed with all the content present within the database server (represented in figure 1). The user would then be able to choose a country/city from the given table (via the shown multi-selector) to either find out the current time of the country/city or the currency conversion of the country/city (chosen by the shown radio buttons).

The currency of the chosen country/city would be displayed in comparison with the current New Zealand dollar. The timezone conversion also would simply tell the user the difference in hours between the chosen country/city and the current New Zealand time, instead of an actual timezone conversion.

VM Purpose and Functionality:

Within this project three virtual machines are used to implement the timezone and currency converter application, those being; a website server VM, database server VM and a VM that manages the conversion. The website server displays the information contained in the database server on a website, along with a multi-selector and radio buttons to allow the user to decide what conversion is going to occur. The purpose of the conversion VM is display either the currency conversion or the timezone conversion for the chosen country/city.

The web server and database server interact directly in order to display the information present within the database onto the website server. The website server and conversion VM also directly interact, allowing the conversion VM to gain the user specifics from the website server, as the user interacts with the website to tell the conversion VM what country/city and conversion type they want displayed. The database server and conversion VM do not directly interact.

These three VM's are used separately to ensure robustness and protection of information. If these three VM's were combined into a single VM, the data and overall program is more at risk to a system failure. Having these separate VM's ensures that if the website server has a system failure the information present within the database is not lost and the methods present in the conversion VM are conserved.

These three VM's being separated also means that they are less susceptible to a security breach. If one VM's information/program was released or targeted maliciously, and if the correct security processes were in place, it would have little effect on the other two VM's individual information/program. Where the project currently stands this breach would still affect the system overall, though if developments were made and back up machines containing duplicates of the current VM's were used, further breaches of security would have no to little effect on the project overall.