PGPCC | Project

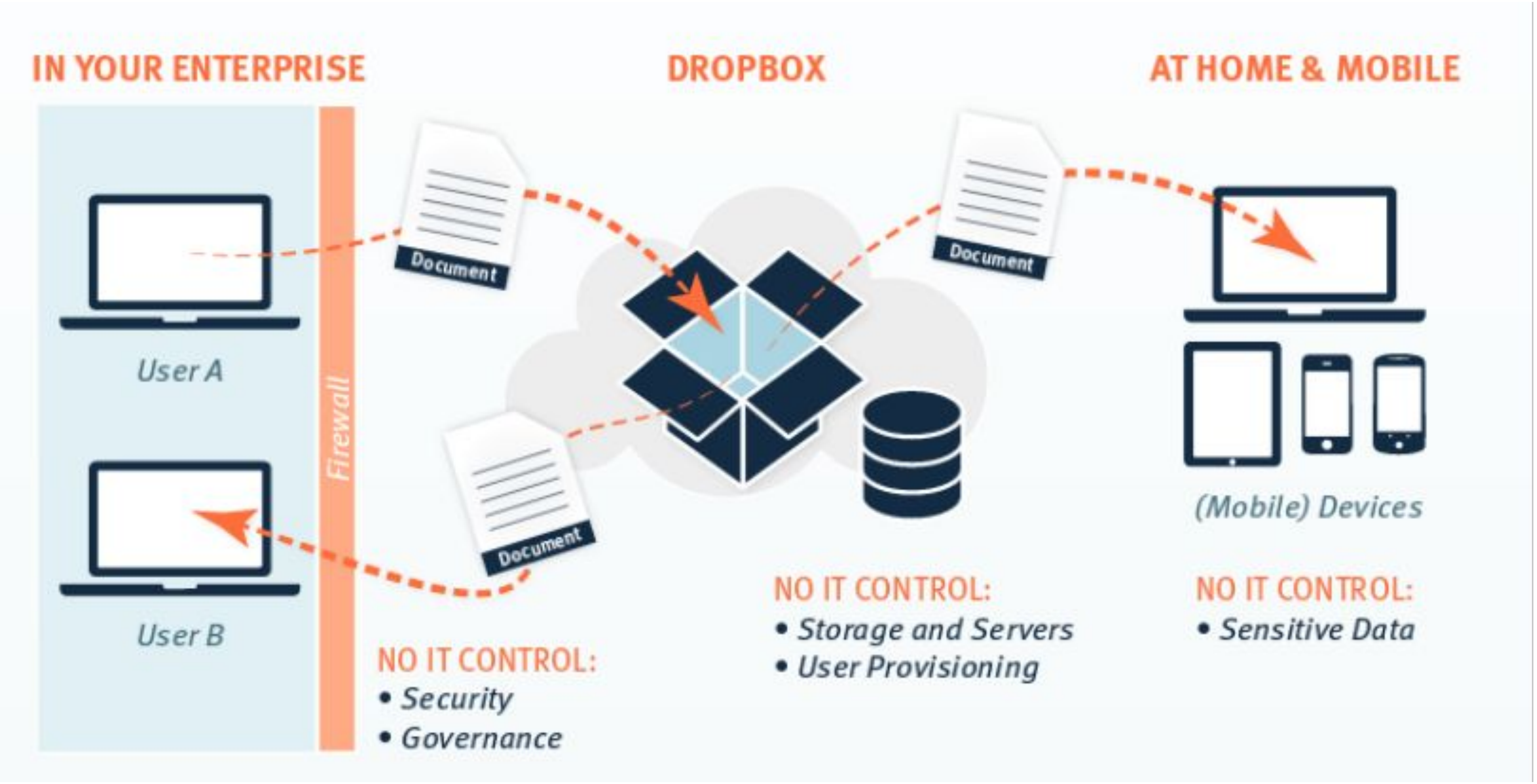
Creating a file share & sync solution using ownCloud and AWS

Scenario – Solving the Dropbox Problem

According to recent research, 40-75% of employees are using Dropbox to share files inside and outside of their businesses. Half of those Dropbox users do this even though they know it's against the rules. More than 40% of businesses have experienced the exposure of confidential information and the estimated average cost of a data breach equaled $5.5 Million in 2011.

These files, containing sensitive company and customer data, are stored in a public cloud outside of the businesses' control - possibly even outside of the country. The potential for data leakage and security breaches is enormous and companies need to stay compliant with their own policies and procedures for security and governance

The Dropbox Problem in Action



References: owncloud.org

The Solution

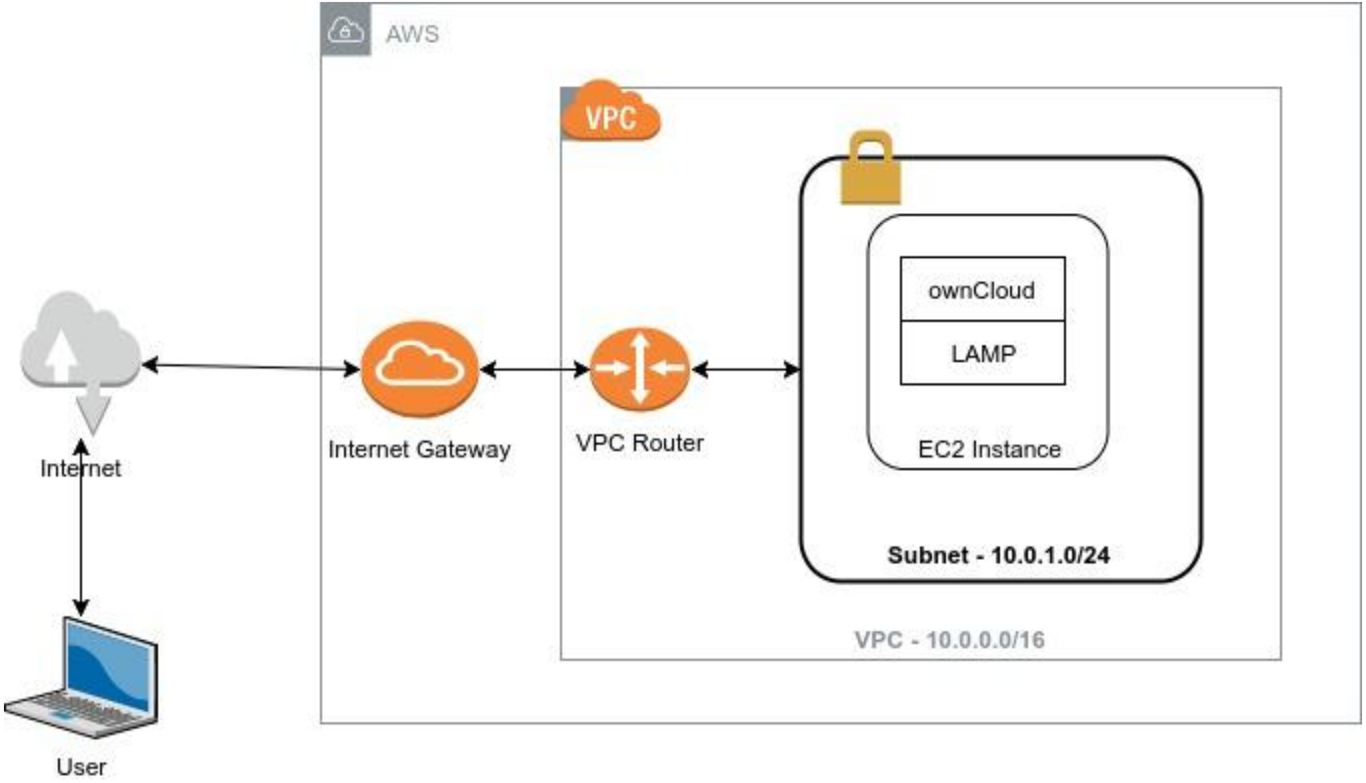
OwnCloud is an open source secure file sync and share solution which can help you gain control of this situation and enable you to create and deploy an enterprise scale file solution. ownCloud can run in your data center or on a public cloud, with its servers, storage etc completely managed and controlled by your IT team and management in accordance with your company's governance and security requirements.

You will implement the ownCloud solution for a small workgroup, which can cater upto 150 users by using various AWS services. Your solution will be completely deployed on public cloud.

References: owncloud.org

Base Architecture

greatlearning



What are you expected to do?

1. **Phase 1 – Architecture**
   Create an architecture diagram for the final implementation (Point 2 in this slide). (hint. will be an update to base architecture). *This is for your implementation reference and you are required to submit this as part of your Solution Document (Project Report).*

2. **Phase 2 – Implementation**

   A. **Implement 2 different subnets** (one public and the other private) **in a custom VPC** called *owncloud-vpc*.

   B. Install and configure MySQL database to run on the private subnet. This subnet should be associated with a security group that allows traffic to private subnet only from the public subnet.

   C. The ownCloud app should be installed in public subnet and MUST be configured to access a new database called *owncloud-db* (created by you) in the private subnet. Apache HTTP server should host ownCloud application in this subnet and must be configured with required PHP modules for ownCloud.

Note - Educate account access policy restrict access to NAT Gateway, use NAT Instance instead

Grading Policy & Tasks

Phase 1 (Architecture) = 10 points
Phase 2 (Implementation) = 70 points
Complete Solution Document = 10 points

Total Project Score = 90 points

Install Apache and PHP on Ubuntu 18.04

1. Create Ubuntu 18.04 instance using 7 steps workflow. Open ports 80 and 22 using security group
2. ssh to created instance
3. Install apache web server using following commands
   1. sudo apt-get update
   2. sudo apt-get install apache2
4. Validate installation by accessing public ip of EC2 instance in browser
5. Use the following commands to install php – sudo apt install php libapache2-mod-php php-mysql
6. Make index.php as the default first load page
   1. Edit /etc/apache2/mods-enabled/dir.conf file and make index.php as first access page
      DirectoryIndex **index.php** index.html index.cgi index.pl index.xhtml index.htm
   2. Restart the web server – sudo systemctl restart apache2

Install OwnCloud on Ubuntu 18.04

1. Run following commands
   1. curl https://download.owncloud.org/download/repositories/10.0/Ubuntu_18.04/Release.key | sudo apt-key add -
   2. echo 'deb http://download.owncloud.org/download/repositories/10.0/Ubuntu_18.04/ /' | sudo tee /etc/apt/sources.list.d/owncloud.list
   3. sudo apt update
   4. sudo apt install php-bz2 php-curl php-gd php-imagick php-intl php-mbstring php-xml php-zip owncloud-files
2. Change default site directory to owncloud files directory using sudo user
   1. edit /etc/apache2/sites-enabled/000-default.conf
   2. update directory root path to /var/www/owncloud
   3. restart the server - sudo systemctl reload apache2
3. Access the owncloud application using public ip of EC2 instance in browser

Install & Configure MySQL on DB server

```
sudo apt-get install mysql-server
sudo mysql_secure_installation
sudo mysql
```

Create new DB and user with all privileges

```
CREATE DATABASE owncloud
CREATE USER ‘owncloud’ @ ‘localhost’ IDENTIFIED BY ‘password’ ;
CREATE USER ‘owncloud’ @ ‘%’ IDENTIFIED BY ‘password’ ;
GRANT ALL PRIVILEGES ON *.* to owncloud@localhost IDENTIFIED BY ‘password’ WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* to owncloud@’%’ IDENTIFIED BY ‘password’ WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

Change the bind address

```
sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
sudo systemctl restart mysql
```

On web server install mysql client

```
mysql -uowncloud -h<ip> -p<password>
```

Helpful Links

- How to install LAMP stack on Ubuntu 18.04
- How To Install and Configure ownCloud on Ubuntu 18.04
- How to Install MySQL DB on EC2 instance
- Owncloud documentation
- Cloud Computing on AWS course – Instructional Modules

Resource CleanUp

1. Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you’ve completed your lab or project is always necessary. This is true whether you’re doing a lab or implementing a project at your workplace.

2. After completing with the lab, make sure to delete each resource created in the reverse chronological order.

Submission Guidelines

- The solution document should strictly follow the sequence of steps listed in “The Solution” slide.
- Each screenshot needs to be qualified with a brief description of what is it about.
- Participants should explicitly write comments and remarks if they wish to notify the evaluator of specific points.
- It is mandatory to share “Lessons & Observations” at the end of the solution document.
- DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.

Academic Honesty & Anti-Plagiarism Policy

Cheating, plagiarism, and all forms of academic dishonesty are expressly forbidden in this program, and by our Policy on Academic Integrity, any form of cheating will immediately earn you a failing grade for the entire course.

Note: Unlike labs where we encourage peer to peer learning and support, participants are strongly advised to not help each other for Projects. This is an individual exercise. Any form of help/support whether offered explicitly, proactively or as a response will be treated as plagiarism.

How to submit your solution?

1. Navigate to the relevant “PROJECTS” course in Olympus.
2. Name your solution document appropriately in the format of: BATCH_FIRSTNAME_LASTNAME_PROJECT1;
   - e.g. PGPCCMAY18_VIJAY_DWIVEDI_PROJECT1.pdf
   - e.g. pgpccmay18_vijay_dwivedi_project1.pdf
3. Upload your solution document and hit submit.
4. Upload any associated files, if you wish to substantiate your solution.

***Note**: If you wish to make modifications to your submitted solution, you can resubmit your solution document “within the submission window” and mark your comments accordingly.*