**Magento Overview**

Technology Stack

PHP

Coding Style (PSR-0, PSR-1, PSR-2, PSR-3 & PSR-4)

Composer

HTML

CSS

jQuery

RequireJS

Third-party libraries (Zend Framework, Symfony)

Apache or Nginx

MySQL

Magento Testing Framework (MTF)

**Architectural layers**

Magento can be divided into four layers

Presentation Shown in browser. It contains layout, blocks, and templates and even

controllers. Client side technology such as jQuery, RequireJS, CSS and

LESS are part of this layer

Services Presentation layer call Service layer

Domain service layer is bridge between domain and presentation layer.

Domain layer implement the business logic.

Persistence this is database layer

**Top-level filesystem structure**

app/etc/di.xml this the most import file, map various classes

var/magento/language-\* all languages are register here. Each module can declare own

translation under /app/code/{vendorName}/{ModuleName}/i18n/

bin in this directory we can find Magento files. Magento files are script

which we can run for different tasks.

dev in this directory test script are there.

lib this directory contain two more directories **internal** (contain PHP code

and fonts) & **web** (contain client side JavaScript)

pub in this directory publically expose files are located. This is the directory

set as root. pub/index.html is the start point of website.

var in this directory dynamically generated file are there , like log, cache etc

vendor this the directory where most of code is located. In this directory all

third-party vendor code, magento modules , themes etc

vendor/magento magento code can be found here in this directory.

**Module filesystem structure**

Magento is modular platform. We will exam magento built-in module call Contact.

It will be located in vendor/magento/module-contact directory

block in this directory where view related PHP classes are located

controller in this directory where view related PHP classes are located

etc in this directory module configuration are present

helper in this directory various helper classes are present

i18n in this directory transaction CSV files are present.

module in this directory entities, resource entities, collections are present

test in this directory module unit test are present.

view store front templates files & static file are present here

**Managing The Environment**

**Setting Up Development Environment**

VirtualBox download the VirtualBox and install on the system

Vagrant download the vagrant and install on the system.

Vagrant Project

Provision PHP

Provision MySQL

Provision Apache

Provision Magento Installation

**Magento 2.3.1 Requirements**

Operating System Ubuntu

Memory 2GB

Composer latest version

Web Server Apache 2.4 (mod\_wirte & mod\_version must enable)

Nginx 1.x

Database MySQL 5.6, 5.7 or MariaDB 10.0, 10.1 & 10.2

PHP PHP 7.2.11

PHP Extensions check magento site for list

PHP OPcache must be enabled

PHP Settings

SSL

System Dependency check magento site for list

Mail Server SMTP

Technology used Redis , Varnish , Elasticsearch, RabbitMQ

Optional / recom php\_xdebug 2.5.x on development system, mcrpt and PHPUnit 6.2.0

**Procedure**

Create a folder with name and location (/Users/rizwan/www/B05032-Magento-Box)

🡪 This is the directory we will pull the magento code

🡪 Magento source code will be external to Vagrant box, so we can work with it in IDE

🡪 Create a vagrant project directory (/Users/rizwan/www/magento-box)

🡪 inside magento-box directory 🡪 run the vagrant init command

🡪 edit the VagrantFile and add the follow code.

🡪 config.vm.box = "bento/ubuntu-18.04"

🡪 config.vm.network "private\_network", ip: "192.168.10.10"

🡪 config.vm.synced\_folder "../B05032-Magento-Box", "/vagrant-B05032-Magento-Box"

🡪 vb.memory = "2028"

🡪 vb.gui = false

🡪 config.vm.provision "file", source: "~/.gitconfig", destination: ".gitconfig"

🡪 config.vm.provision "shell", inline: "sudo apt-get update"

🡪 install git in HOST system and configure globally the username and email

🡪 git config --global user.name “Rizwan Khan”

🡪git config --global user.email [rizwan.wakil2018@gmail.com](mailto:rizwan.wakil2018@gmail.com)

🡪 Provisioning PHP

🡪

**Setting Up Production Environment**

Amazon Web Services register for AWS

Setting up for S3 usage

Creating IAM User

Creating IAM Group

Setting up S3 for database and media files backup

Bash script for automated EC2 Setup

Setting up EC2 Instance

Setting up Elastic IP & DNS