

Full marks - 100 marks

Project marks - 75 marks

MCQ marks- 25 marks

Deadline- 8th Aug 9AM

Submission link -

https://docs.google.com/forms/d/e/1FAIpQLSf6rnZZJiqD7CsNjVvjCadcrKYJZwYowjLHCOvO5NcvUw51Yw/viewform?usp=sf_link

Part-1 SDLC: Scrum 30 marks

Overview :

Everyday we come across different problems, situations and we come up with ideas to solve them. You have to select one of the ideas yourself and describe the following requirements of the product development lifecycle.

1. Identify the problems. Write the description of your app, software or website in 200 words or less. **5 marks**

Last year our group in the university had to do an application project which was an app when person enter it can detect their face we did call it facial recognition app We as group used a milestone plan to finish the application.
Our scrum master was our adviser

1 concepts

- Owner of the app was looking for an app to get exact needs of it
- Product owner discussed the requirement
- He also was discussing the features for app such as it can detect the person if they are laughing or sad

2 inception

As a small team we can not chose specific developers to work on the project so We divided the work for each team member as what they can do

- One is doing the documentation
- Second is doing the digram and sequins digram for each face of the app
- Third is doing the coding part

3 lietration

Is time for the real work

- each of the team member has to deliver thair work on time
- This stage developers are in need for the scrum master mpre then anyb

time

4 release

- A developer while they are working in app or site they should test every step they go through
- Project was delivered and release after all needed tests are done

For us we did not deploy it at that time because it needs ongoing support to make it run as it needs

2. Describe the roles and responsibilities in your project for the following roles. **5 marks**

2.1. Scrum Master

- Scrum master
Is the one who is taking care for the project and should be stick to the plan for arranging meetings and keep team on track
We used to have our adviser having meeting for us every Friday for 15 minutes to discuss and share our progress

2.2. Product Owner

- Product owner
Should have knowledge of the market strategy and show their vision and take the lead of many areas of the product

2.3. Developers

- As project developers work actively on each phase of the application
- Make our responsibility complete on time
- Do the documentation

3. Simulate the scrum events for your project and fill the table below with details on a simulated sprint. **5 marks**

Events	Items planned in bullet points.
Sprint Planning	<ul style="list-style-type: none">● An event that make the team start of plan and discuss their plan
Daily Scrum	<ul style="list-style-type: none">● Short daily meeting to discuss the team work and plan
Sprint Review	<ul style="list-style-type: none">● Fast review for our hard work for designing and developing and so
Sprint Retrospective	<ul style="list-style-type: none">● This is to share what we were succeed with during last sprint and what can be improved
Sprint	<ul style="list-style-type: none">● For coding my team mate was having problems so we started to do the coding in parallel of time to get on track

4. Write the definition of done for a sprint just before the first public launch of the software. **5 marks**

- For our work we are exactly finish within 2 weeks before the release date which was our plan to
- Even checking list of all done work

5. Write the list of items in following sprint artifacts. **10 marks**

5.1. Product Backlog

- New features to make the project needed in the market
- Updates to make it work smoothly
- Bug fixing to have user trust the product

5.2. Sprint Backlog

- Sprint backlogs are made by choosing a task from the product backlog and subdividing it into smaller tasks

5.3. Done Increment

- Done Increment All Scrum Teams must at the very least abide by it because it is a requirement of the organization.

Part-2: Ansible

1. Ansible basics. 20 marks

Write a playbook adhering to the following requirements.

- 1.1. Create an inventory file with **AmzLinuxservers** and **Ubuntuservers** groups. Add one server in each of these groups.

Paste the inventory file. 5 marks


Vi hosts

```
ubuntu@ip-172-31-83-121:~/ansible-workspace$ cat hosts
[amzlinuxserver]
ec2-user@ec2-18-208-110-248.compute-1.amazonaws.com

[ubuntuservers]
ec2-184-73-147-63.compute-1.amazonaws.com
ubuntu@ip-172-31-83-121:~/ansible-workspace$
```

- 1.2. Create a playbook to perform the following tasks.
 - 1.2.1. install apache on both the groups.
 - 1.2.2. Create or copy **index.html** with following content on both the groups.
<h1>Hello **YOUR-NAME**</h1>
 - 1.2.3. Install and start apache on both the groups.

Paste the playbook file you wrote. 5 marks



Ubuntu

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- *-parts.conf
|-- mods-enabled
|   |-- *-load
|   |-- *-conf
|-- conf-enabled
|   |-- *-conf
|-- sites-enabled
|   |-- *-conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.

Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.


If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting `www.example.com`, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf/4welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

2.4

```

ubuntu@ip-172-31-83-121:~/ansible-workspace$ cat copy.yml
---
- hosts: ec2-user@ec2-18-208-110-248.compute-1.amazonaws.com
  become: true
  tasks:
    - name: Install apache in Amazon Linux servers amzlinuxservers
      yum:
        name:
          - httpd
        state: present
        update_cache: yes

    - name: starting apache
      service: name=httpd state=started enabled=yes
    - name: Set up HTML file
      copy: src=/home/ubuntu/ansible-workspace/index.html dest=/var/www/html/index.html

- hosts: ubuntu servers
  become: true
  tasks:
    - name: Install apache in ubuntu servers Ubuntu servers
      apt:
        name:
          - apache2
          - php
        state: present
        update_cache: yes
    - name: Ensure apache starts
      service: name=apache2 state=started enabled=yes
    - name: Set up HTML file
      ansible.builtin.copy: src=/home/ubuntu/ansible-workspace/index.html dest=/var/www/html/index.html
ubuntu@ip-172-31-83-121:~/ansible-workspace$

```

```

ubuntu@ip-172-31-83-121:~/ansible-workspace$ cat index.html
<html>
<h1>welcom Bader Alghamdi</h1>
</html>

ubuntu@ip-172-31-83-121:~/ansible-workspace$

```

- 1.3. Paste the screenshot of both the servers with their ip. It should be showing your name **Hello YOUR-NAME**

Write the command you used to run the playbook. **5 marks**

```

ubuntu@ip-172-31-83-121:~/ansible-workspace$ ansible-playbook install-apache.yml
PLAY [Install Apache on Web server] *****
TASK [Gathering Facts] *****
ok [ec2-18-208-110-248.compute-1.amazonaws.com]
TASK [Install Apache server] *****
ok [ec2-18-208-110-248.compute-1.amazonaws.com]
TASK [Start apache server] *****
ok [ec2-18-208-110-248.compute-1.amazonaws.com]
TASK [Done] *****
ok [ec2-18-208-110-248.compute-1.amazonaws.com] FAILED! => "changed" state, "yes" does not exist!
PLAY RECAP *****
ec2-18-208-110-248.compute-1.amazonaws.com : ok=3  changed=0  unreachable=0  failed=1  skipped=0  rescued=0  ignored=0

ubuntu@ip-172-31-83-121:~/ansible-workspace$ vi hosts
ubuntu@ip-172-31-83-121:~/ansible-workspace$ vi install-apache.yml
ubuntu@ip-172-31-83-121:~/ansible-workspace$ ansible-playbook install-apache.yml

```

```
ubuntu@ip-172-31-83-121:~/ansible-workspace$ cat index.html
<html>
<h1>welcom Bader Alghamdi</h1>
</html>

ubuntu@ip-172-31-83-121:~/ansible-workspace$
```

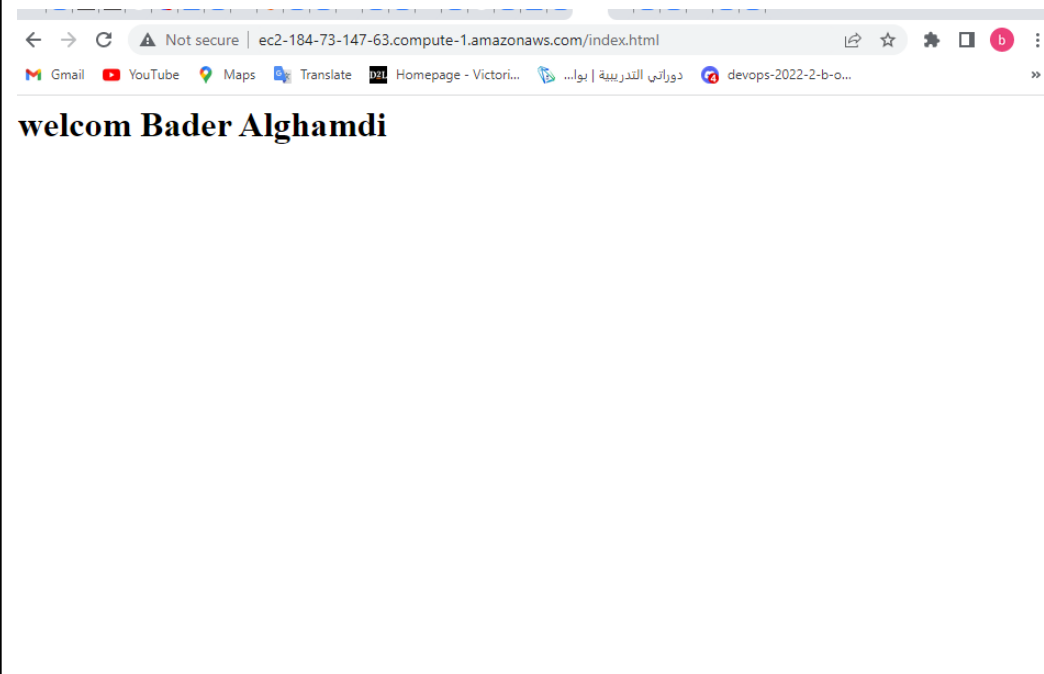
Screenshot: AmzLinuxserver with IP address. **2.5 marks**

← → ↻ Not secure | ec2-18-208-110-248.compute-1.amazonaws.com

Gmail YouTube Maps Translate Homepage - Victori... دوراني التدريبية | بوا... devops-2022-2-b-o... Learn - Student Das... 3y4ag2f37 - JavaScr... LMS | Whizlabs

welcom Bader Alghamdi

Screenshot: Ubuntu server with IP address. **2.5 marks**



2. Ansible advanced. 25 marks

- 2.1. Create an ansible vault using `ansible-vault secret.yml` and store a username and password in it.

For example:

username: YOUR-NAME (*without space*)
password: @sdf98w4tlkgf (*anything*)

10 marks

Write the content of secret.yml file

```
ubuntu@ip-172-31-83-121: ~
```

```
username: bader-alghamdi  
password: 12345
```

Write the command to create the secret

```
ubuntu@ip-172-31-83-121:~$ ansible-vault create secret.yml
```

Show the screenshot of the `ansible-vault view secret.yml` command

```
ubuntu@ip-172-31-83-121:~$ ansible-vault create secret.yml
New Vault password:
Confirm New Vault password:
ubuntu@ip-172-31-83-121:~$ ansible-vault view secret.yml
Vault password:
username: bader-alghamdi
password: 12345
ubuntu@ip-172-31-83-121:~$
```

ubuntu@ip-172-31-83-121:~\$ `ansible-vault view secret.yml`

- 2.2. Write a playbook to create user `create-user.yml` it should be using the username and password created in the last `secret.yml` ansible vault.
15 marks

Paste the content of `create-user.yml`

```
ubuntu@ip-172-31-83-121:~$ cat create-user.yml
---
- name: create a user
  hosts: all
  become: true
  vars_files:
    - secret.yml
  tasks:
    - name: creating user
      user:
        name: "{{ username }}"
        password: "{{ password }}"
ubuntu@ip-172-31-83-121:~$
```

Write the command to create user using `create-user.yml` and with same inventory file

```
password: [! password ?]
ubuntu@ip-172-31-83-121:~$ sudo ansible-playbook --ask-vault-pass create-user.yml
Vault password:
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'

PLAY [create a user] *****
skipping: no hosts matched

PLAY RECAP *****

ubuntu@ip-172-31-83-121:~$ sudo ansible-playbook --ask-vault-pass create-user.yml -i hosts
Vault password:
[WARNING]: * Failed to parse /home/ubuntu/hosts with yaml plugin: We were unable to read either as JSON nor YAML,
these are the errors we got from each: JSON: Expecting value: line 1 column 2 (char 1) Syntax Error while loading
YAML. did not find expected <document start> The error appears to be in '/home/ubuntu/hosts': line 2, column 1,
but may be elsewhere in the file depending on the exact syntax problem. The offending line appears to be:
[webserver] web2 ansible host=ec2-184-73-147-63.compute-1.amazonaws.com ansible user=ubuntu ansible ssh
pass=Pass@123 ^ here
[WARNING]: * Failed to parse /home/ubuntu/hosts with ini plugin: /home/ubuntu/hosts:2: Expected key=value host
variable assignment, got: ansible
[WARNING]: Unable to parse /home/ubuntu/hosts as an inventory source
[WARNING]: No inventory was parsed, only implicit localhost is available
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not
match 'all'

PLAY [create a user] *****
skipping: no hosts matched

PLAY RECAP *****

ubuntu@ip-172-31-83-121:~$
```

Verify the user created or not in the group nodes. Screenshot of the output of the command to show users created in the inventory or not.

The command could be similar to this:

ansible all -b -m command -a "grep YOUR-USERNAME /etc/shadow" -i inventory

```
ubuntu@ip-172-31-83-121:~$ sudo ansible all -b -m command -a "grep bader-alghamdi/etc/shadow"
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'
ubuntu@ip-172-31-83-121:~$
```