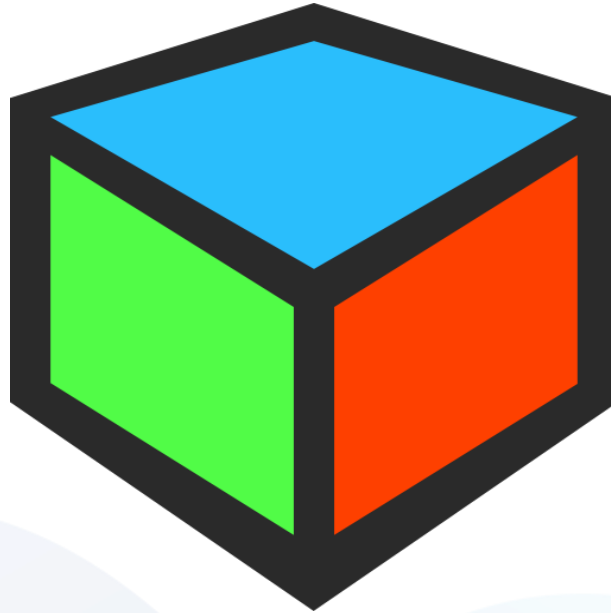


ORCHESTRATION AND HARDENING AUTOMATION OF LAMP STACK ON GOOGLE CLOUD PLATFORM

Abdullah Alhuwaish, Bandar Almutairi

SACRED HEART UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING





SIMPLICITY

PRODUCTIVELY

OPEN SOURCE





Agenda

- Project Timeline
- Implementation Tasks & milestones





Project Timeline

Estimated duration
In Weeks

01

Start 9/11/2018

End 9/18/2018

1

Research design & planning

- Finalise research problem/questions
- Develop research design
- Prepare research proposal

02

Start 9/11/2018

End 10/2/2018

3

Literature review

- Search, capture and synthesise relevant literature
- Prepare draft literature review

03

Start 10/2/2018

End 10/23/2018

3

Data collection

- Finalise sampling plan
- Develop data collection instrument
- Pre-test/pilot data collection instrument
- Carry out data collection
- Write up data collection

04

Start 10/23/2018

End 11/6/2018

2

Data analysis

- Prepare data for analysis
- Analyze data
- Draw conclusions/ recommendations

05

Start 11/6/2018

End 11/25/2018

1.8

Writing up

- Final draft of report
- Review draft with supervisor
- Final editing
- Printing, binding and final submission





Implementation Tasks & milestones

09/1
09118

1



ANSIBLE

Ansible

- Ansible and Ansible Architecture.
- Ansible mechanics
- Learn YAML language which is used for ansible configuration.





MILESTONES ONE

YAML

An organized unit of scripts that defines work for a server configuration managed by Ansible



ANSIBLE

Playbooks

Templates

Modules

File which contains all your configuration parameters

Modules can control system resources, like services, packages, or handle executing system commands





MILESTONES ONE (Cont.)

Use Ansible
Galaxy
Structure

```
1. bandar@Bandar: ~/Desktop/Final Project /Code of Project/playbook_ssh (zsh)
bandar@Bandar ~/Desktop/Final Project /Code of Project/playbook_ssh tree
.
├── hosts
├── roles
│   └── ssh
│       ├── README.md
│       ├── defaults
│       │   └── main.yml
│       ├── files
│       ├── handlers
│       │   └── main.yml
│       ├── meta
│       │   └── main.yml
│       ├── tasks
│       │   └── main.yml
│       ├── templates
│       │   └── sshd_config.j2
│       ├── tests
│       │   ├── inventory
│       │   └── test.yml
│       └── vars
│           └── main.yml
├── setup.sh
└── ssh.yml

10 directories, 12 files
bandar@Bandar ~/Desktop/Final Project /Code of Project/playbook_ssh
```





Implementation Tasks & milestones

09/1
09118

1



ANSIBLE

Ansible

- Ansible and Ansible Architecture.
- Ansible mechanics
- Learn YAML language which is used for ansible configuration.

09/1
10802

2



SSH

- SSH configuration
- Hardening SSH configuration
- Write YAML file to automate SSH hardening
- key distribution





MILESTONES TWO

**Shall Script
to install
Packages on
GCP**

```
1  sudo apt-get -y install python
2  sudo apt-get -y update
3  sudo apt-get -y install python-pip python-dev build-essential
4  sudo apt-get -y update
5  sudo pip install --upgrade pip
6  pip install --upgrade virtualenv
7  sudo apt-get -y install software-properties-common
8  sudo apt-add-repository ppa:ansible/ansible -y
9  sudo apt-get -y update
10 sudo apt-get -y install ansible
11 sudo pip install apache-libcloud
```

**RSA key
2048 bit and
use GCP
metadata**

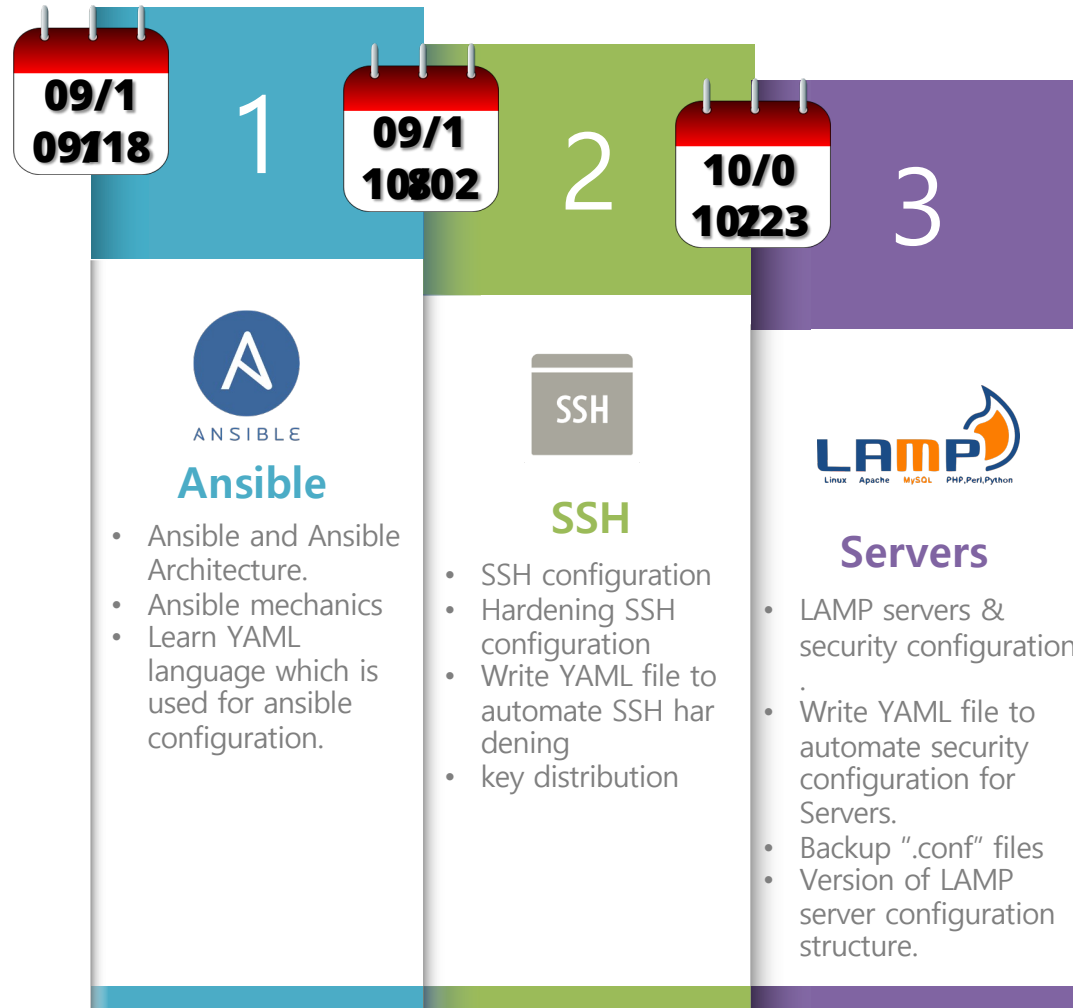


**KEY
DISTRIBUTION**





Implementation Tasks & milestones





MILESTONES THREE

Apache Script Tasks

```
1  ---
2  - name: i just want to install apache
3    apt: pkg=apache2 state=latest update_cache=true
4
5  - name: install php
6    ignore_errors: yes
7    apt: pkg=php state=latest update_cache=true
8
9  - name: install php-mysql
10   ignore_errors: yes
11   apt: pkg=php-mysql state=latest update_cache=true
12
13  - name: load config file
14    template: src=dir.conf.j2 dest=/etc/apache2/mods-enabled/dir.conf
15
16  - name: load config file
17    template: src=index.php.j2 dest=/var/www/html/index.php
18
19  - name: start apache service
20    service: name=apache2 state=restarted
```





MILESTONES THREE

Load Balancer Script Tasks

```
1  ---
2  - name: i just want to install haproxy
3    become: yes
4    apt: pkg=haproxy state=latest update_cache=true
5
6  - name: copy config file 1
7    template: src=haproxy.j2 dest=/etc/default/haproxy
8
9  - name: copy config file 2
10   template: src=haproxy.cfg.j2 dest=/etc/haproxy/haproxy.cfg
11
12   notify:
13     - restart haproxy
```





MILESTONES THREE

SQL Script Tasks

```
1 ---
2 - name: Installing python-mysqldb package
3   become: yes
4   apt: pkg=python-mysqldb state=latest update_cache=true
5
6 - name: Installing MariaDB
7   become: yes
8   apt: pkg=mariadb-server state=latest update_cache=true
9   notify:
10     - restart mysql
11
12 - name: change bind mysql remote address
13   ini_file: dest=/etc/mysql/my.cnf
14             section=mysql
15             option=bind-address
16             value={{ ansible_eth0.ipv4.address }}
17   notify:
18     - restart mysql
19
20 - name: Add new user
21   mysql_user:
22     name: world
23     password: vagrant
24     priv: '*.*:ALL'
25     host: '%'
26     state: present
27
28 - name: Copy database world.sql
29   become: yes
30   copy:
31     src: ~/playbooks/roles/sql/templates/contacts.sql
32     dest: /tmp
33 - name: Import world.sql into Database
34   mysql_db:
35     login_user: world
36     login_password: vagrant
37     state: import
38     name: contacts
39     target: /tmp/contacts.sql
```





MILESTONES THREE

SSH Script Tasks

```
1 ---
2 - name: load config file
3   template: src=sshd_config.j2 dest=/etc/ssh/sshd_config
4   notify:
5     - restart ssh
```





MILESTONES THREE

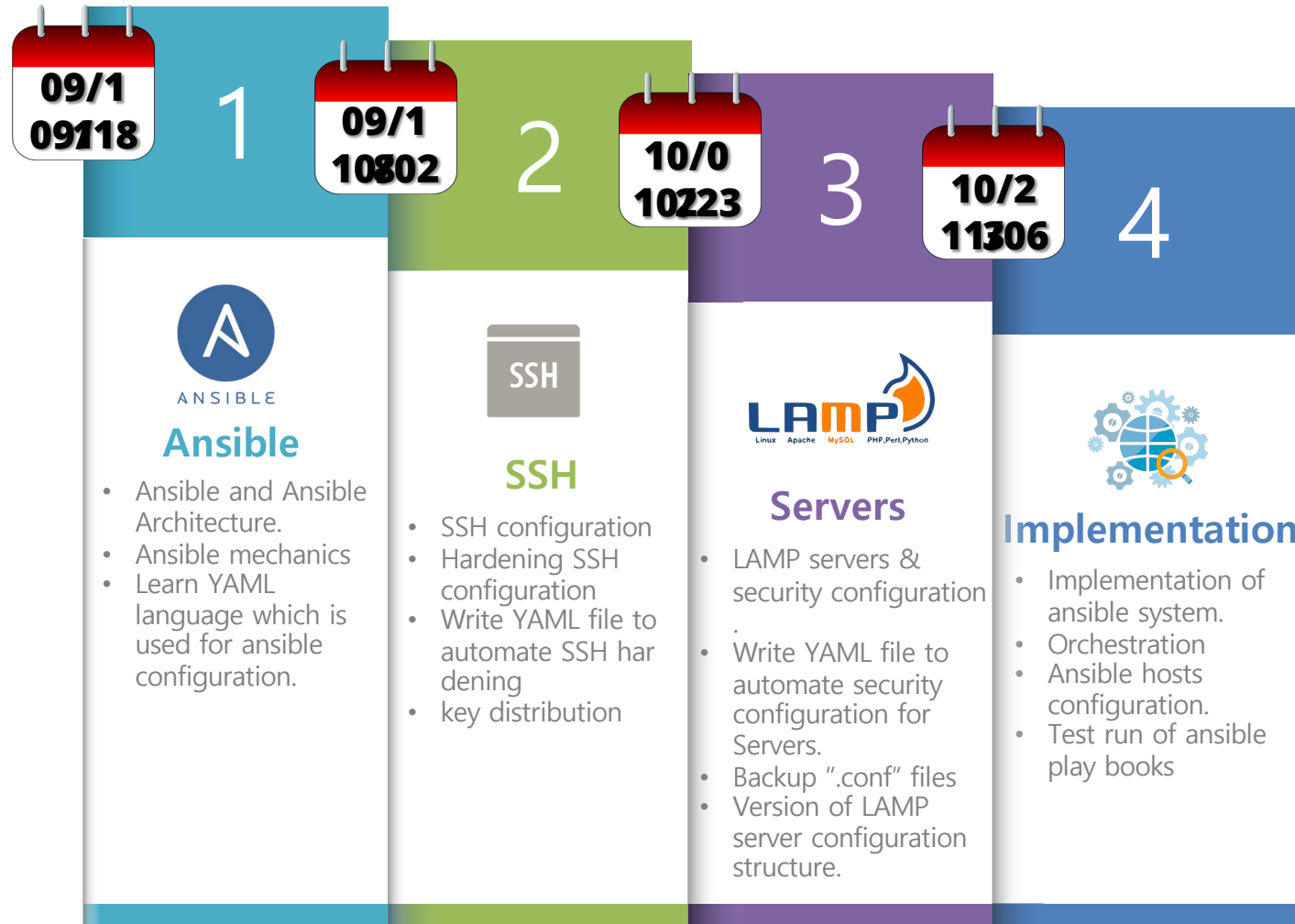
Iptables Script roles

```
1 ---
2 ~ - name: install iptables
3   apt: pkg=iptables-persistent state=latest update_cache=true
4   become: yes
5
6 ~ - iptables:
7   chain: INPUT
8   protocol: tcp
9   destination_port: 22
10  jump: ACCEPT
11
12 ~ - iptables:
13   chain: INPUT
14   protocol: tcp
15   destination_port: 80
16   jump: ACCEPT
17
18 ~ - iptables:
19   chain: INPUT
20   protocol: tcp
21   destination_port: 3306
22   jump: ACCEPT
23 ~ - iptables:
24   chain: INPUT
25   ctstate: ESTABLISHED,RELATED
26   jump: ACCEPT
27
28 ~ - iptables:
29   chain: INPUT
30   jump: DROP
31
32 ~ - iptables:
33   chain: OUTPUT
34   protocol: tcp
35   ctstate: ESTABLISHED,RELATED
36   jump: ACCEPT
37
38 ~ - iptables:
39   chain: OUTPUT
40   jump: ACCEPT
41
42 ~ - iptables:
43   chain: FORWARD
44   jump: DROP
45
46 ~ - name: Saving IPTables Rules using "sudo netfilter-persistent save"
47   ignore_errors: yes
48   become: yes
49   command: "sudo netfilter-persistent save"
```



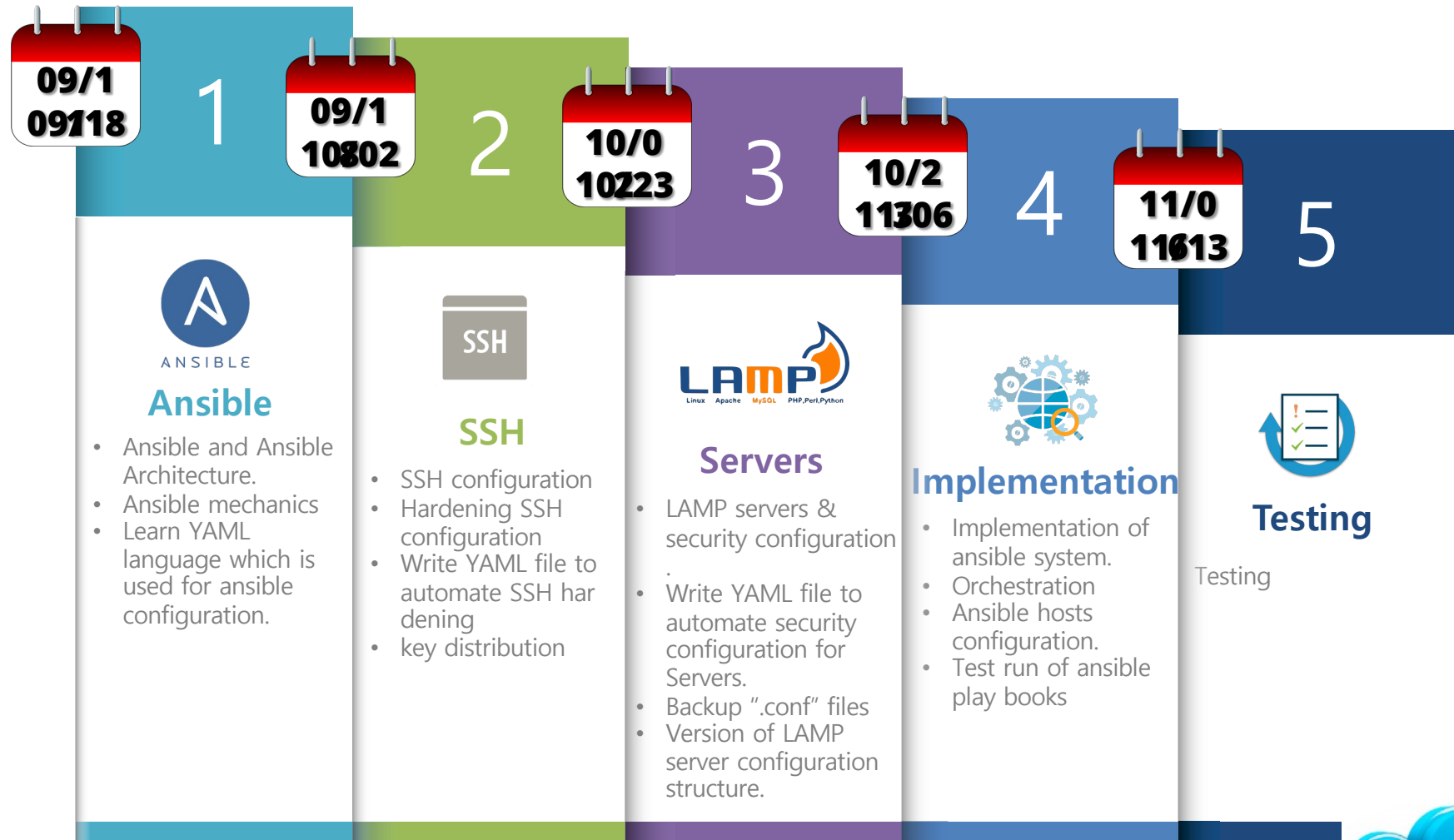


Implementation Tasks & milestones





Implementation Tasks & milestones



THE PROJECT

Implementation

