Infrastructure Automation

Infrastructure Automation

Terraform

ANSIBLE

CHEF

puppet

Ansible

Ansible

- What is Ansible?
  - Ansible is an open source automation engine
- Why Ansible?
  - Modular
  - Idempotent
  - Huge support/community
  - Agentless
  - Simple to ~~learn~~ *start learning*

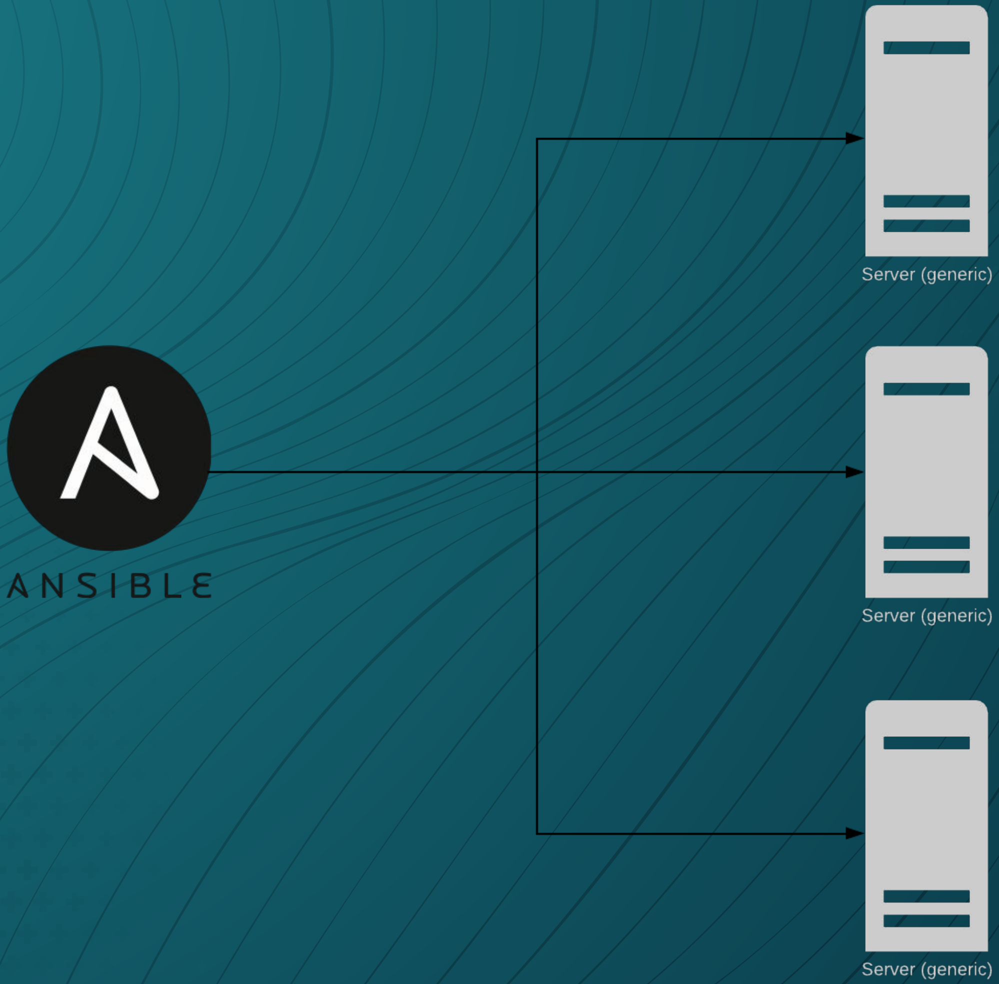
Where and who uses Ansible

- DevOps (Ex. Deploying a webserver)
- Network Engineers (Ex. managing firewalls, switches)
- Security Engineers (Ex. blocking malicious users)
- Systems Engineers (Ex. managing active directory)

- We (lockdown, labs, etc.)

What can ansible manage?

- Linux (SSH)
- Windows (WinRM)
- Firewalls (SSH)
- Switches (SSH)
- Cloud platforms like AWS, Azure, vCenter, etc. (API)
- Itself
- More


ANSIBLE
Server (generic)
Server (generic)
Server (generic)

Main components of Ansible

- Inventory (INI, YAML, Python-JSON)
- Modules (YAML)
- Playbook (YAML)

YAML

Example of a **list** in yaml:

```
- Wake up
- Brush your teeth
- Eat breakfast
- Go to school
- Lunch
- Come back home
- Do homework
- Go to sleep
```

Example of a map (**dictionary**) in yaml:

```
city:
  Name: Buffalo
  Area: 52.5 sq mi
  Population: 261,310
  Rank by population in NY: 2
  Timezone: UTC–05:00 (EST)
```

Combining Both:

```
-  aibek:
     name: Aibek
     job: Student
     skills:
       - Python
       - Ansible
       - Windows
-  stephen:
     name: Stephen
     job: Developer at Google
     skills:
       - Golang
       - Python
       - Linux
```

Note: dictionaries are represented in a “key: value” format.
Examples of keys above: Name, Area, etc.
Examples of values above: Buffalo, 52.5 sq mi, etc.
More info on YAML: Ansible Yaml Documentation

Playbooks

```
- name: Network Getting Started First Playbook Extended      # name of a “Play”
  hosts: 192.168.5.20                        # hosts to which the “Play” will be applied
  tasks:                               # tasks that will be applied to the host

    - name: Ping a device                    # name of task 1
      ping:                            # module for task 1

    - name: Creates directory                # name of task 2
      file:                            # module for task 2
        path: /src/www                       # parameter for module in task 2
        state: directory                     # parameter for module in task 2
```

Note: “#” represent comments that could be written directly into YAML

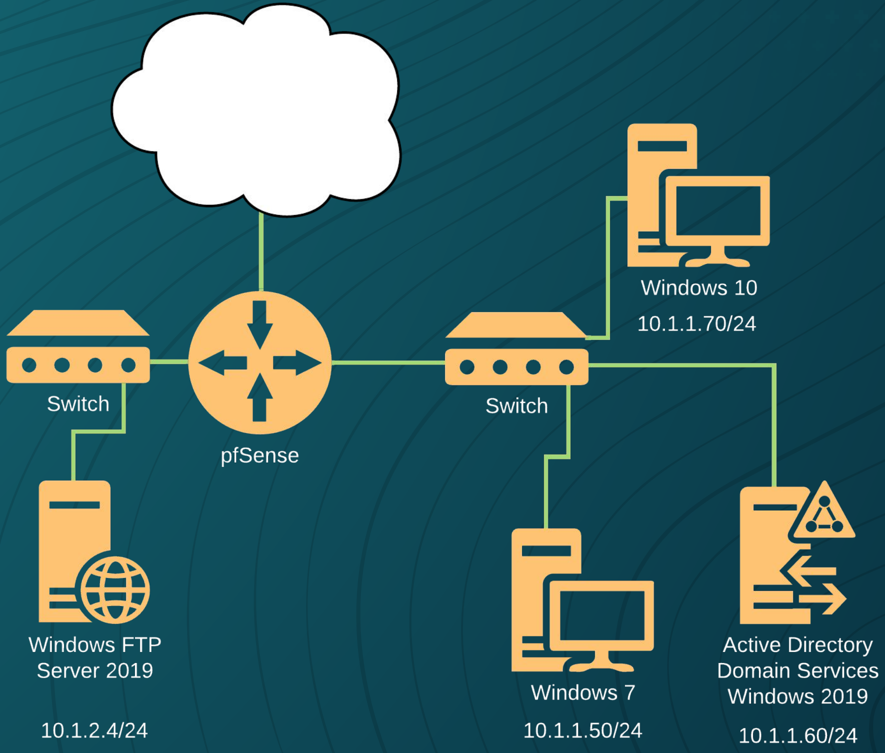
Inventory Files (Hosts File) .INI

```
[windowsclients]  # Name of the group
10.1.1.70         # Member of the group
10.1.1.50

[windowsftp]
10.1.2.4

[windowsad]
10.1.1.60

[windows:children]      # "Group of groups"
windowsclients    # Members of a bigger group
windowsad
windowsftp
```



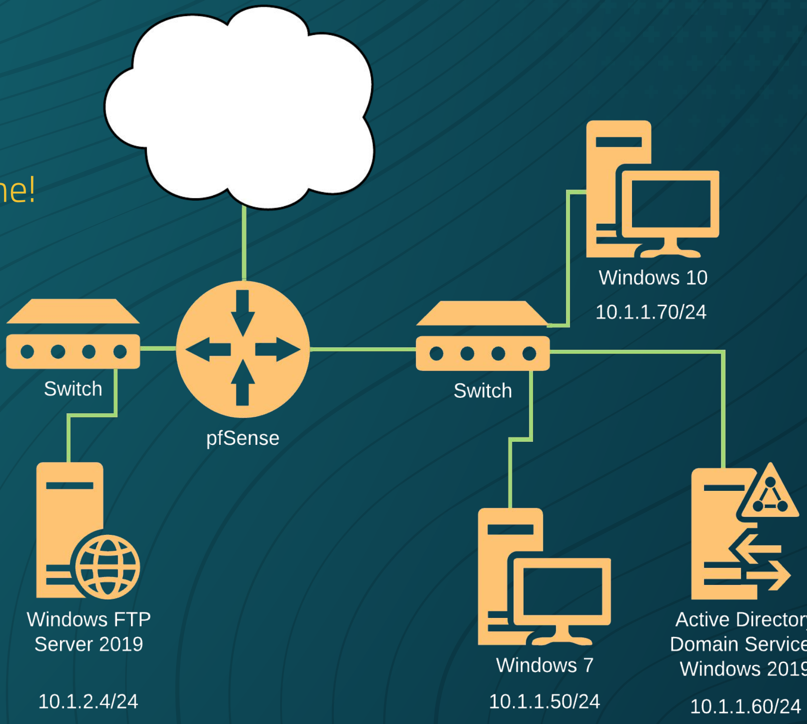
Variables

```
[windowsftp]
10.1.2.4 ansible_user=Administrator ansible_password=Change.me!

[windowsad]
10.1.1.60 ansible_user=Manager ansible_password=Change.me!

[windows:children]
windowsad
windowsftp

[windows:vars]
ansible_connection=winrm
```



Note: Generally, variables could be assigned globally, per group, or per host

Modules

Modules (also referred to as “task plugins” or “library plugins”) are discrete units of code that can be used from the command line or in a playbook task. Ansible executes each module, usually on the remote target node, and collects return values. – Ansible

Modules in a Playbook: Playbooks

Examples of Modules:
file - Manage files and file properties
ping - Try to connect to host, verify a usable python and return pong on success
shell - Execute shell commands on targets
vmware_guest – Manages virtual machines in vCenter

Note: Google and ansible module documentation page are best resources for researching Ansible Modules.

Conditions and loops

```
tasks:
  - name: "shut down Debian flavored systems"
    command: /sbin/shutdown -t now
    when: ansible_facts['os_family'] == "Debian“

 - name: “Ping all RedHat based distros except Fedora"
    ping:
    when:
       - ansible_facts['os_family'] == "RedHat“
       - ansible_facts[‘distribution’] != “Fedora“

 - name: “Create a remote directory on Windows“
   win_file:
     path: C:\Temp
     state: directory
   when: ansible_facts['os_family'] == “Windows"
```

```
tasks:
 - name: Add multiple users
   user:
     name: andrew
 - name: Add multiple users
   user:
     name: shanelle
 - name: Add multiple users
   user:
     name: aritra
```

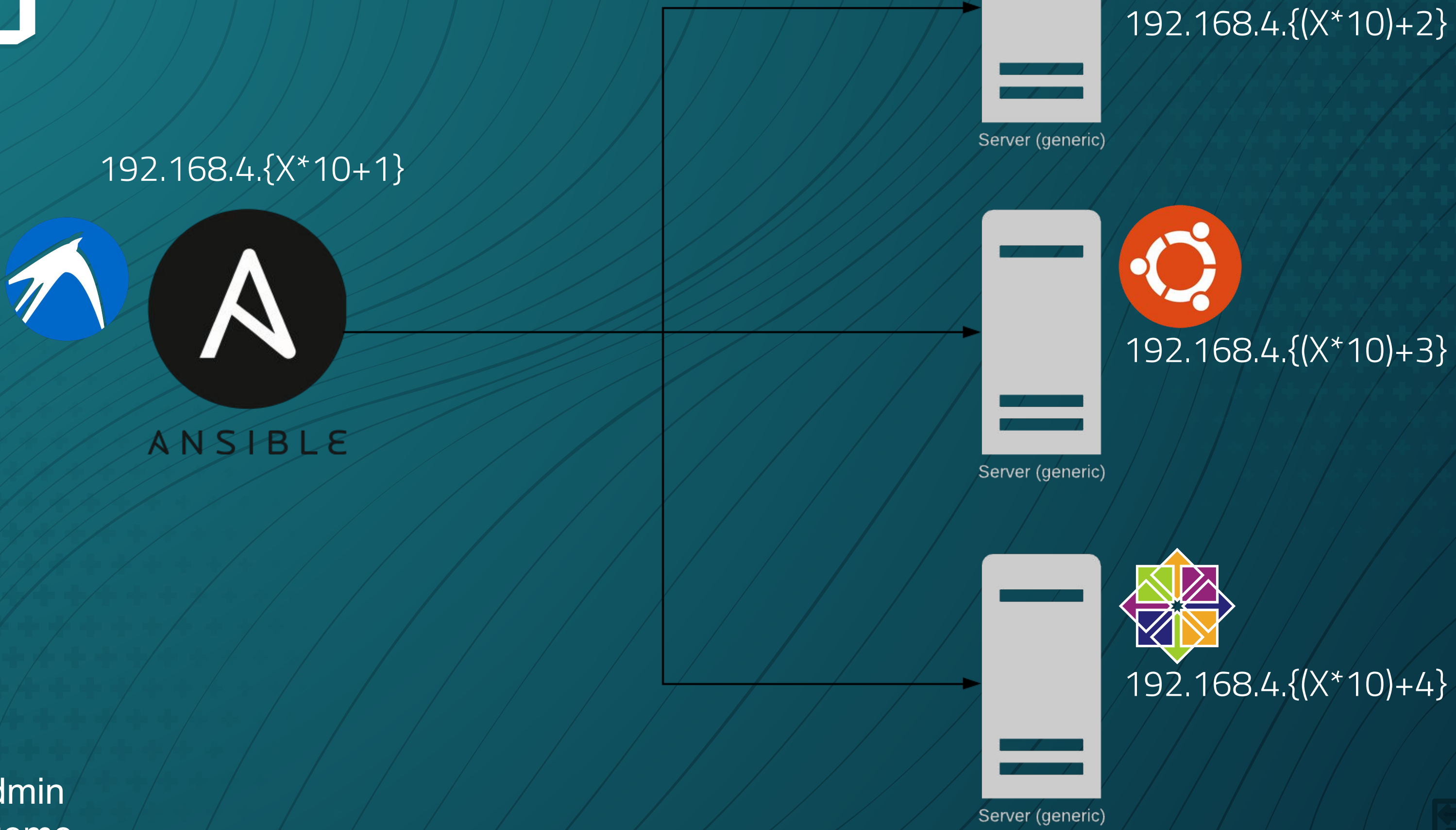
```
tasks:
 - name: Add multiple users
   user:
     name: “{{ item }}”
   with_items:
     - andrew
     - shanelle
     - aritra
```

Ansible Roles

Ansible role is a collection of variables, tasks, files, etc in a specific file structure:

```
./
    inventory.ini                         # Inventory file
    webservers.yml                                 # Playbook
    roles/                                # folder named “role” under which roles are stored
        common/                           # role called common
            tasks/                        # A Collection of tasks,
            files/                        # files
            vars/                         # variables
            defaults/                     # (default variables)
        webservers/
            tasks/
            defaults/
```

LAB



Username: sysadmin
Password: changeme

Objective 1: Ensure Apache is running Centos

1) Configure Inventory
2) Update Remote Servers
3) Install Apache
4) Enable Apache Service (AutoStart)
5) Start Apache Service

Objective 2: Ensure Ubuntu.2 has directory /tmp/2 created and Ubuntu.3 has directory /tmp/3 created

1) Configure Inventory
2) Define variables
3) Create directory

Objective 3: Use existing role from previous Lockdowns to deploy mediawiki website

1) Clone repository [https://bit.ly/32Kp8Rg]
2) Configure inventory Dynamic Inventory
3) Configure playbook
4) Run

Advanced Topics

Objective 4: You choose:

1) Configure Inventory

Facts Gathering

- By default, before starting the task, Ansible will get “facts”, the information about system using setup module.
- The information gathered could be used with Conditionals and loops
- Example of using “facts gathering” in roles: https://github.com/geerlingguy/ansible-role-mysql

Ansible Vault

- Ansible Valut allows you to store sensitive information like passwords in an encrypted format

Excerpt from YAML inventory file that utilizes ansible vault:

```
ansible_user: Admininstrator
          ansible_host: 10.1.1.50
          ansible_connection: winrm
          ansible_password: !vault |
            $ANSIBLE_VAULT;1.1;AES256
346363636430613739343032373134326535633964364430646230376163326438353433366
233353732373637663138223936383865333637343839396531380a36333336323332623962
383864366163666363356330376366663865306539636336386665346461396263383133646
6343639663964346437666234363963320a3165303962663061356664623733646139633962
36616662306132623332 3935
```

Dynamic Inventory

- Dynamic Inventory is an inventory generated by a scripted language, namely Python.
- Example: https://github.com/ubnetdef/Lockdown-v7/blob/master/inventory.py
- Use cases:
  - The hosts are now known before hand
  - Shorten length of inventory file

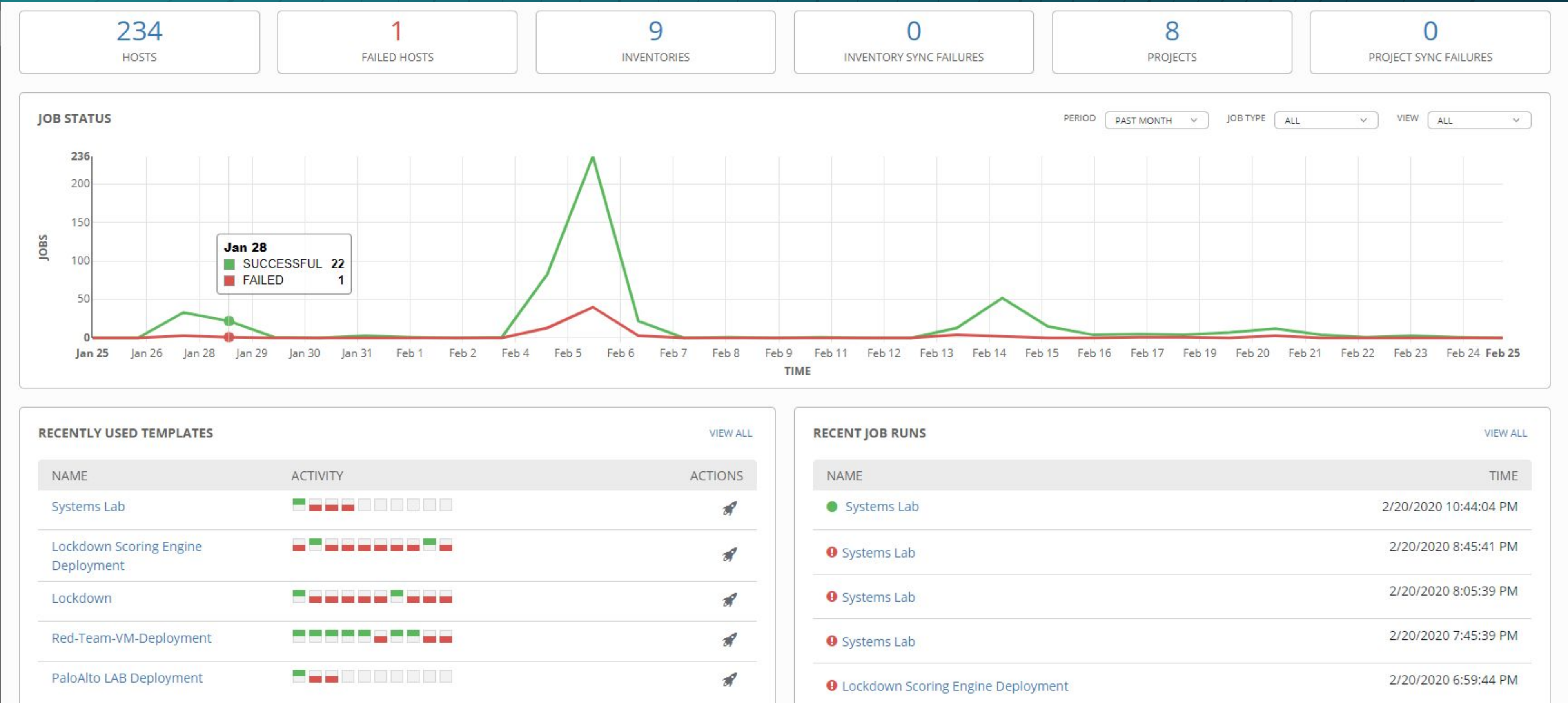
Custom Ansible Modules

- Custom modules are modules created by typical Ansible users using Python

- Making your own module: https://docs.ansible.com/ansible/latest/dev_guide/developing_modules_general.html

Ansible Tower

Lockdown, Labs, and more



234	1	9	0	8	0
HOSTS	FAILED HOSTS	INVENTORIES	INVENTORY SYNC FAILURES	PROJECTS	PROJECT SYNC FAILURES

JOB STATUS

PERIOD: PAST MONTH | JOB TYPE: ALL | VIEW: ALL

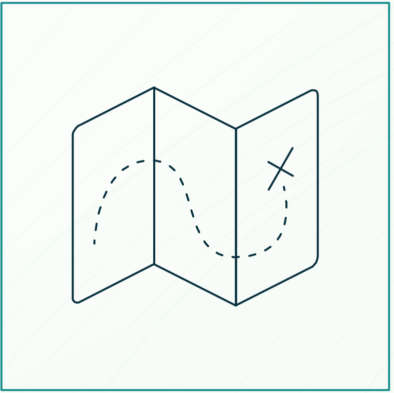
RECENTLY USED TEMPLATES VIEW ALL

NAME	ACTIVITY	ACTIONS
Systems Lab	🟩🟥🟥🟥□□□□□□	🚀
Lockdown Scoring Engine Deployment	🟥🟩🟥🟥🟥🟥🟥🟥🟩🟥	🚀
Lockdown	🟩🟥🟥🟥🟥🟥🟩🟥🟥🟥	🚀
Red-Team-VM-Deployment	🟩🟩🟩🟩🟩🟥🟩🟩🟥🟥	🚀
PaloAlto LAB Deployment	🟩🟥🟥□□□□□□□	🚀

RECENT JOB RUNS VIEW ALL

NAME	TIME
● Systems Lab	2/20/2020 10:44:04 PM
❗ Systems Lab	2/20/2020 8:45:41 PM
❗ Systems Lab	2/20/2020 8:05:39 PM
❗ Systems Lab	2/20/2020 7:45:39 PM
❗ Lockdown Scoring Engine Deployment	2/20/2020 6:59:44 PM

Homework



Thanks!

Any questions?

You can find me at @l1ghtman

Thanks SlidesCarnival for the template!