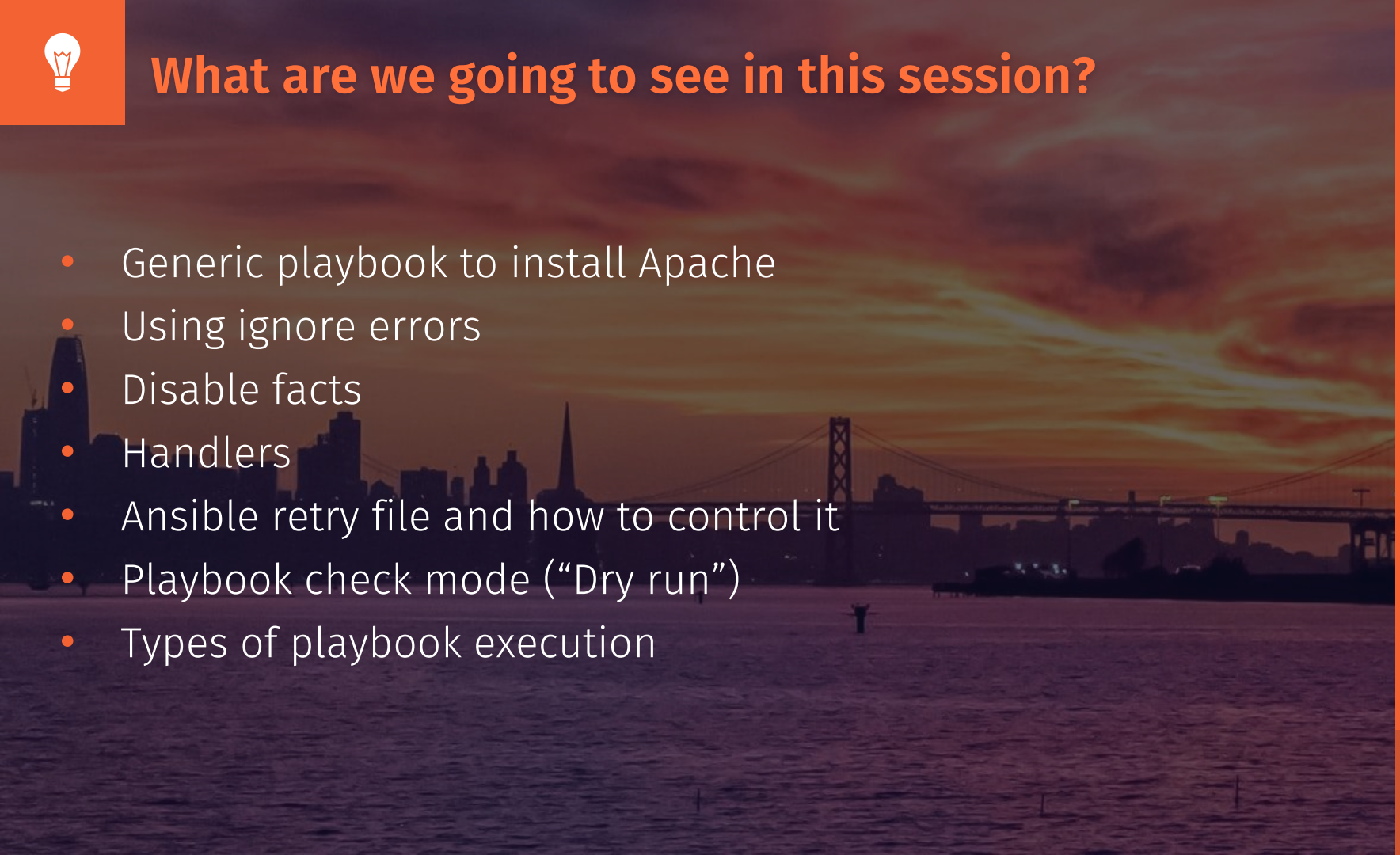


What are we going to see in this session?

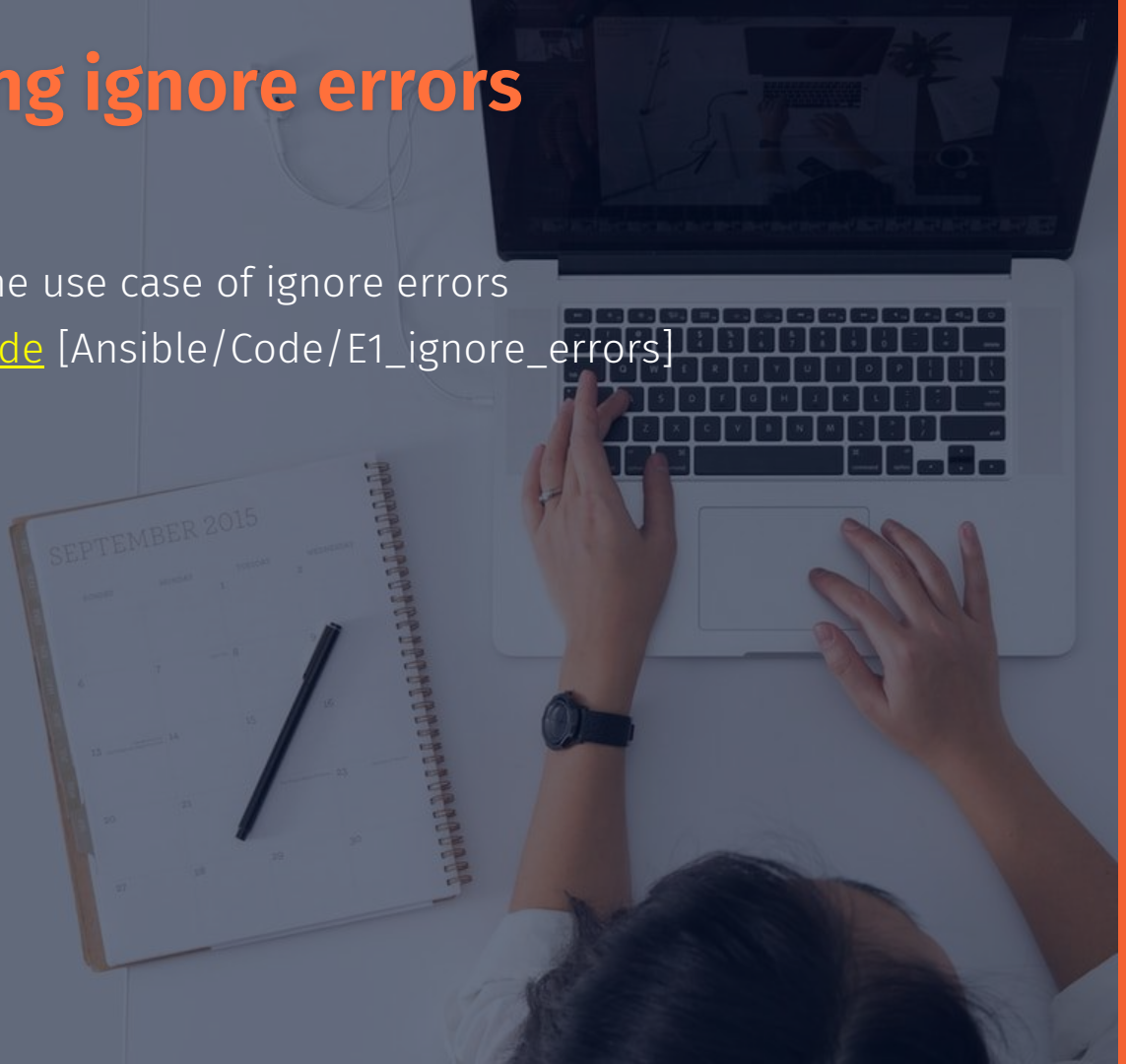
- Generic playbook to install Apache
- Using ignore errors
- Disable facts
- Handlers
- Ansible retry file and how to control it
- Playbook check mode (“Dry run”)
- Types of playbook execution

Generic playbook to install Apache

- Let's see an example here
  GitHub_Code [Ansible/Code/E0_generic]

Using ignore errors

- Let’s see the use case of ignore errors
  GitHub Code [Ansible/Code/E1_ignore_errors]



Disable facts

- You can see every time when you are executing the play it gathers the facts about remote machine by default which also consumes lot of time.
- To avoid such thing you can disable it by adding additional line in playbook.

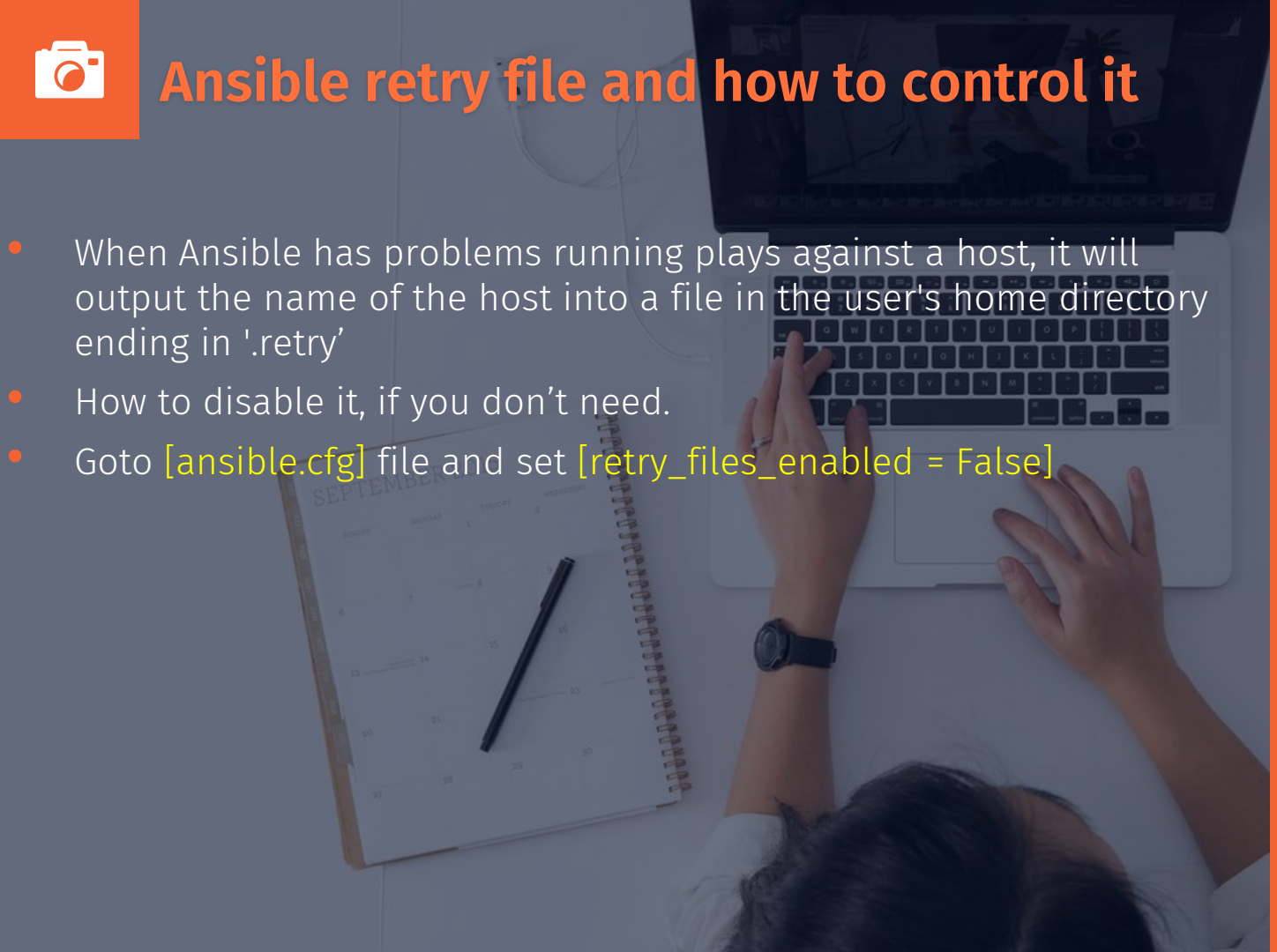
  gather_facts: no
- GitHub_Code [Ansible/Code/E2_facts]

Handlers

- Handlers are lists of tasks, not really anything different from regular tasks.
- These handlers are referenced by an global unique name and are notified by notifiers.
- If nothing notifies a handler, it will not run.
- Regardless of how many tasks notify a handler, it will run only once.
- These ‘notify’ actions are triggered at the end of each block of tasks in a play and will only be triggered once even if notified by multiple different tasks.
- For instance, multiple resources may indicate that apache needs to be restarted because they have changed a config file, but apache will only be bounced once to avoid unnecessary restarts.
- Handlers also will run only at end of the play not before that.
- GitHub Code [Ansible/Code/E3_handlers]

Ansible retry file and how to control it

- When Ansible has problems running plays against a host, it will output the name of the host into a file in the user's home directory ending in '.retry’
- How to disable it, if you don’t need.
- Goto [ansible.cfg] file and set [retry_files_enabled = False]

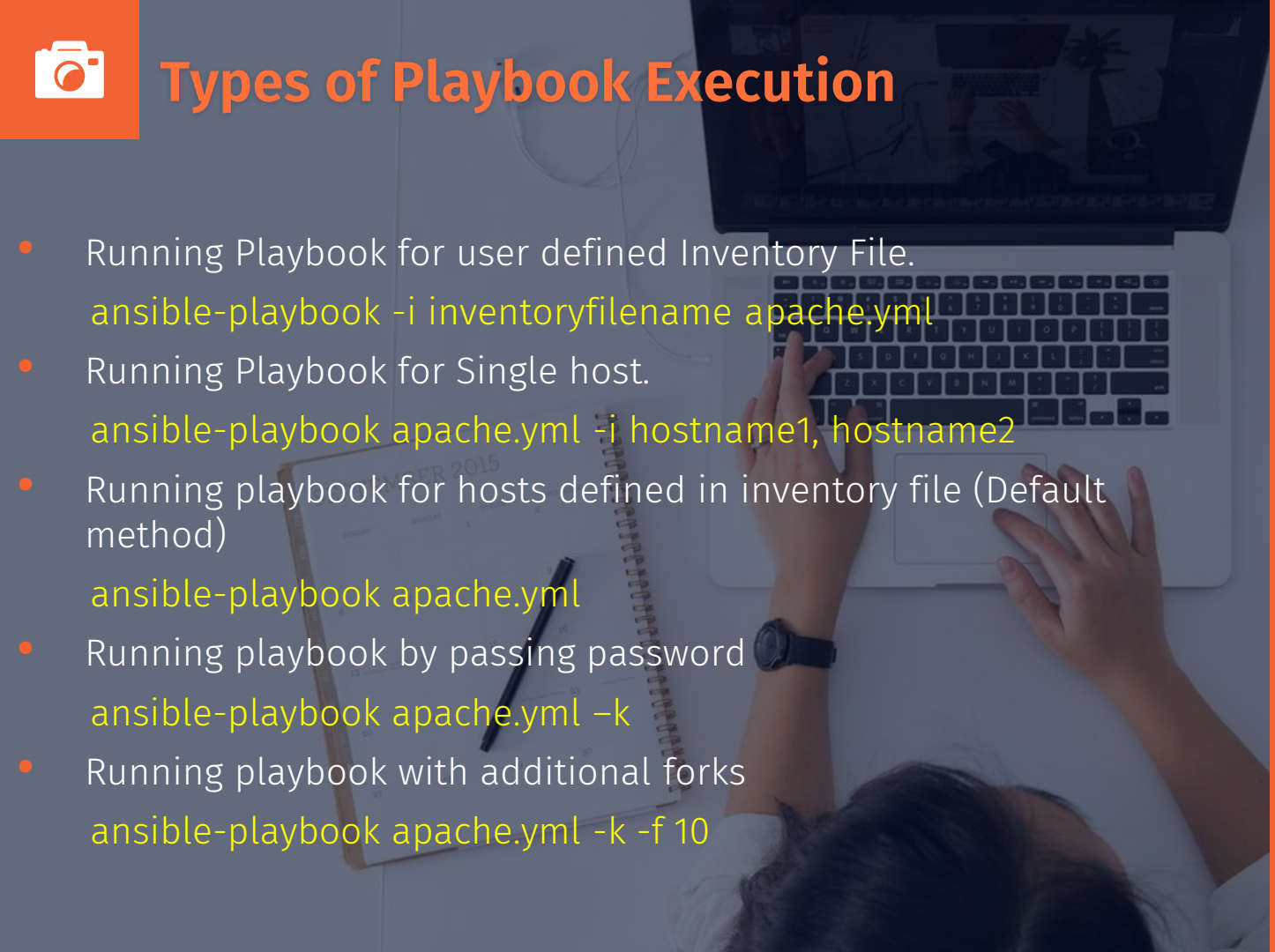


Playbook check mode (“Dry Run”)

- When ansible-playbook is executed with --check it will not make any changes on remote systems.
- ansible-playbook foo.yml –check
- Sometimes you may want to use check mode behavior of individual tasks. This is done via the [check_mode] option, which can be added to tasks.
- There are two options:
  - ✓ Force a task to run in check mode, even when the playbook is called without --check. This is called [check_mode: yes]
  - ✓ Force a task to run in normal mode and make changes to the system, even when the playbook is called with --check. This is called [check_mode: no]

```
tasks:
  - name: this task will make changes to the system even in check mode
    command: /something/to/run --even-in-check-mode
    check_mode: no

  - name: this task will always run under checkmode and not change the system
    lineinfile:
        line: "important config"
        dest: /path/to/myconfig.conf
        state: present
    check_mode: yes
```



Types of Playbook Execution

- Running Playbook for user defined Inventory File.

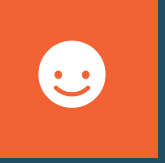
  ansible-playbook -i inventoryfilename apache.yml
- Running Playbook for Single host.

  ansible-playbook apache.yml -i hostname1, hostname2
- Running playbook for hosts defined in inventory file (Default method)

  ansible-playbook apache.yml
- Running playbook by passing password

  ansible-playbook apache.yml –k
- Running playbook with additional forks

  ansible-playbook apache.yml -k -f 10



End of this topic !

Any questions?