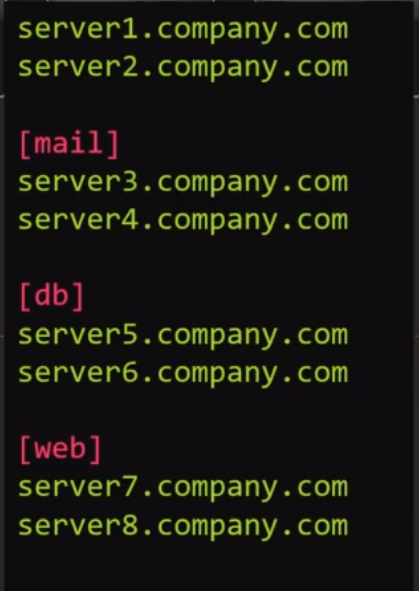
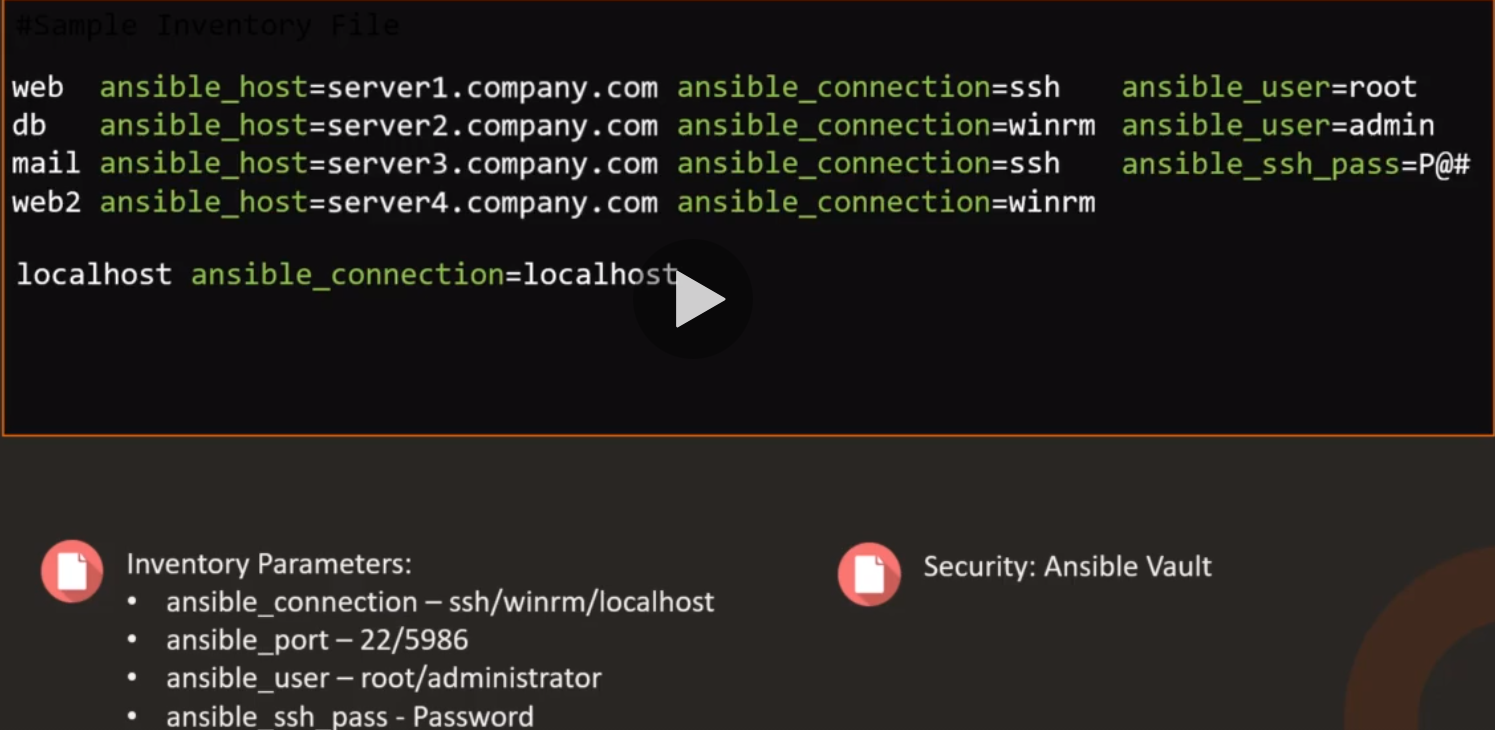
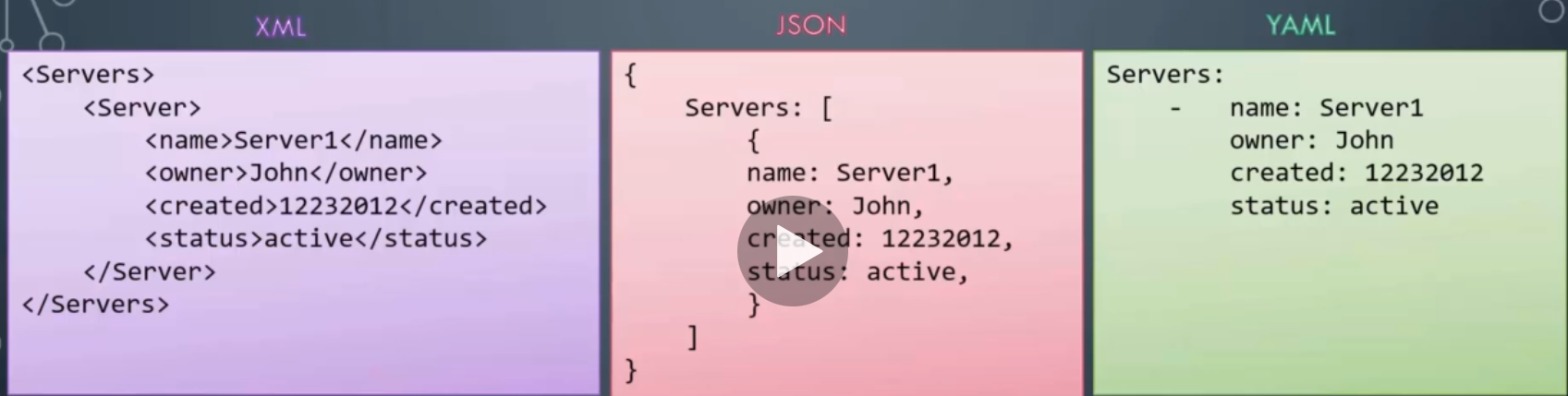
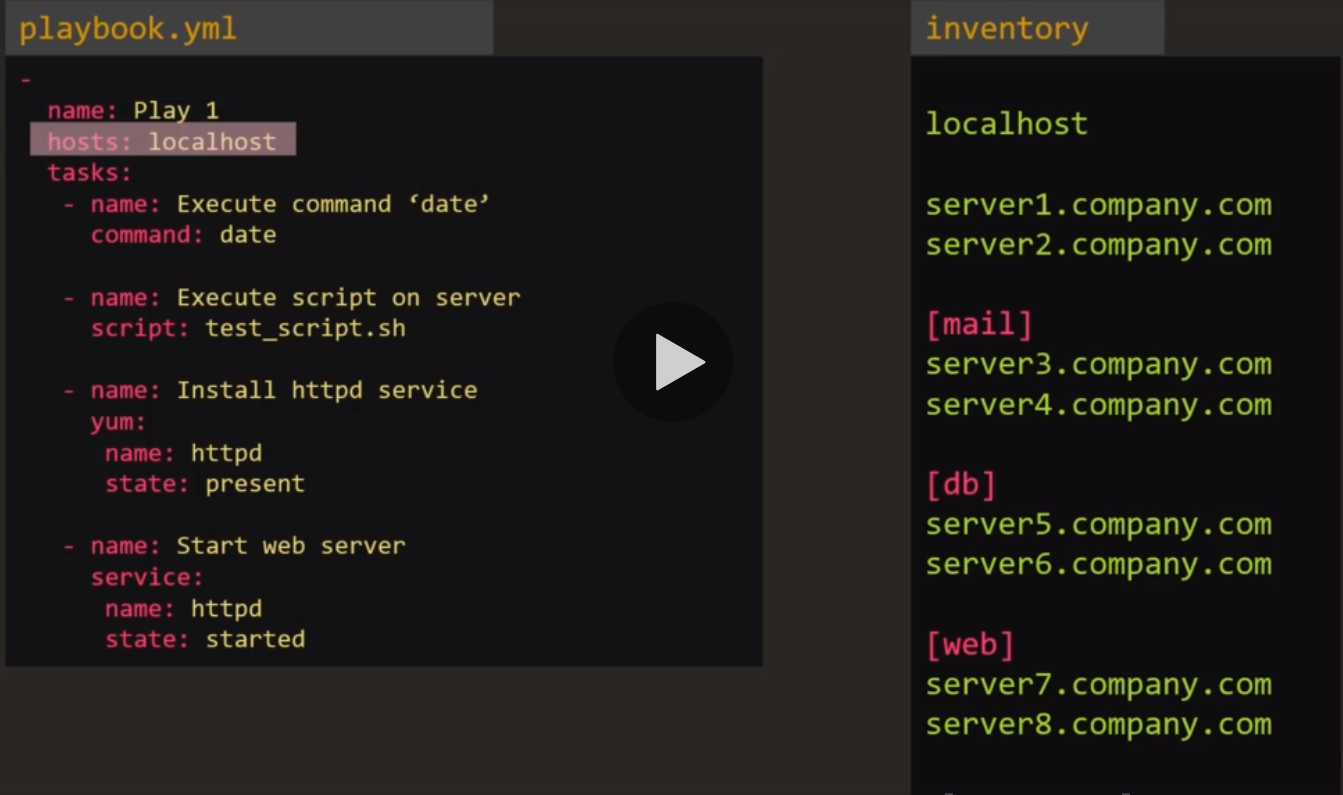
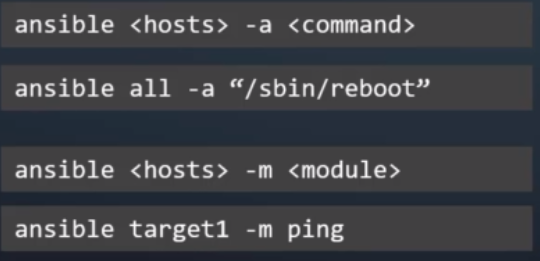
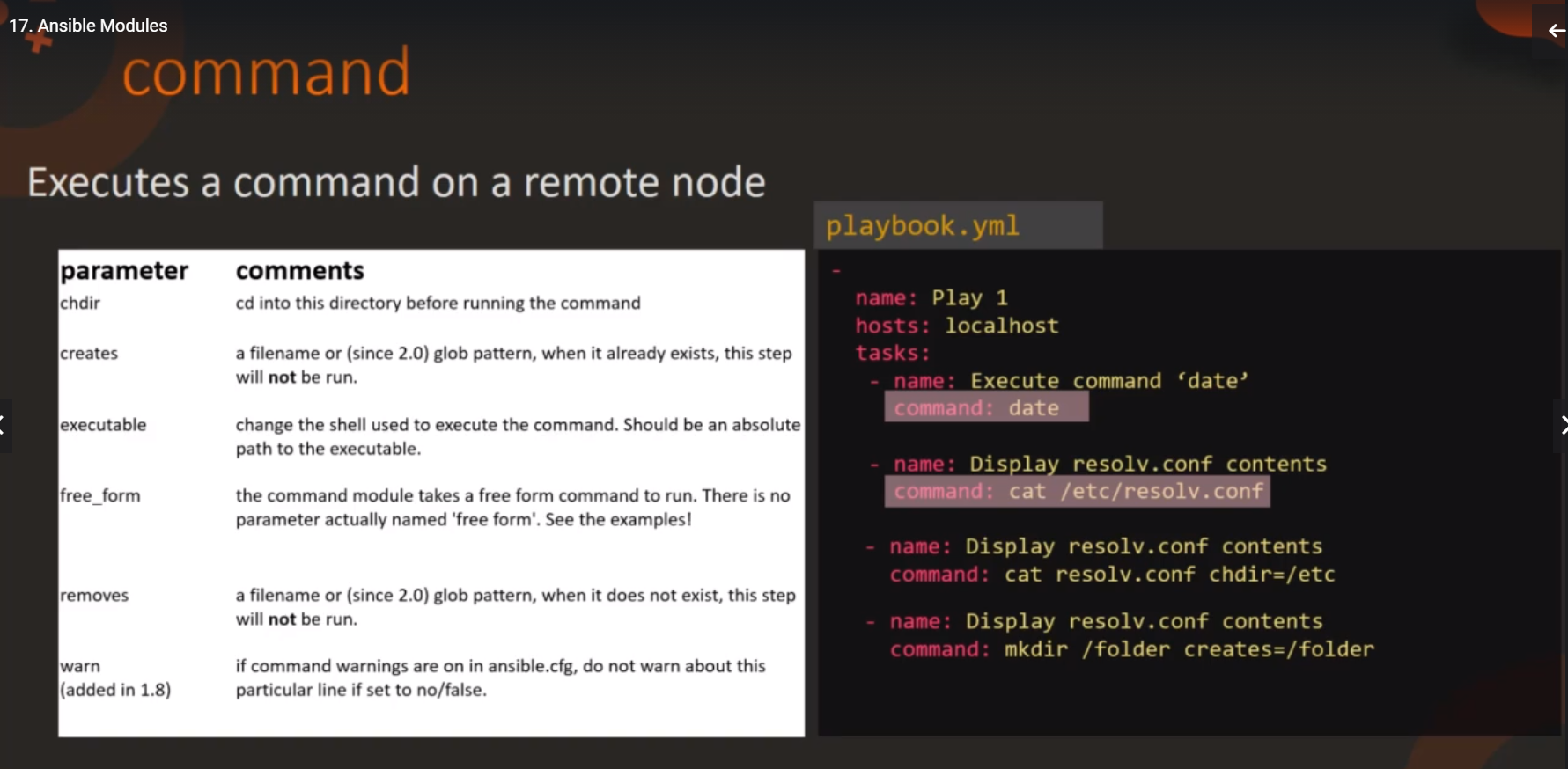
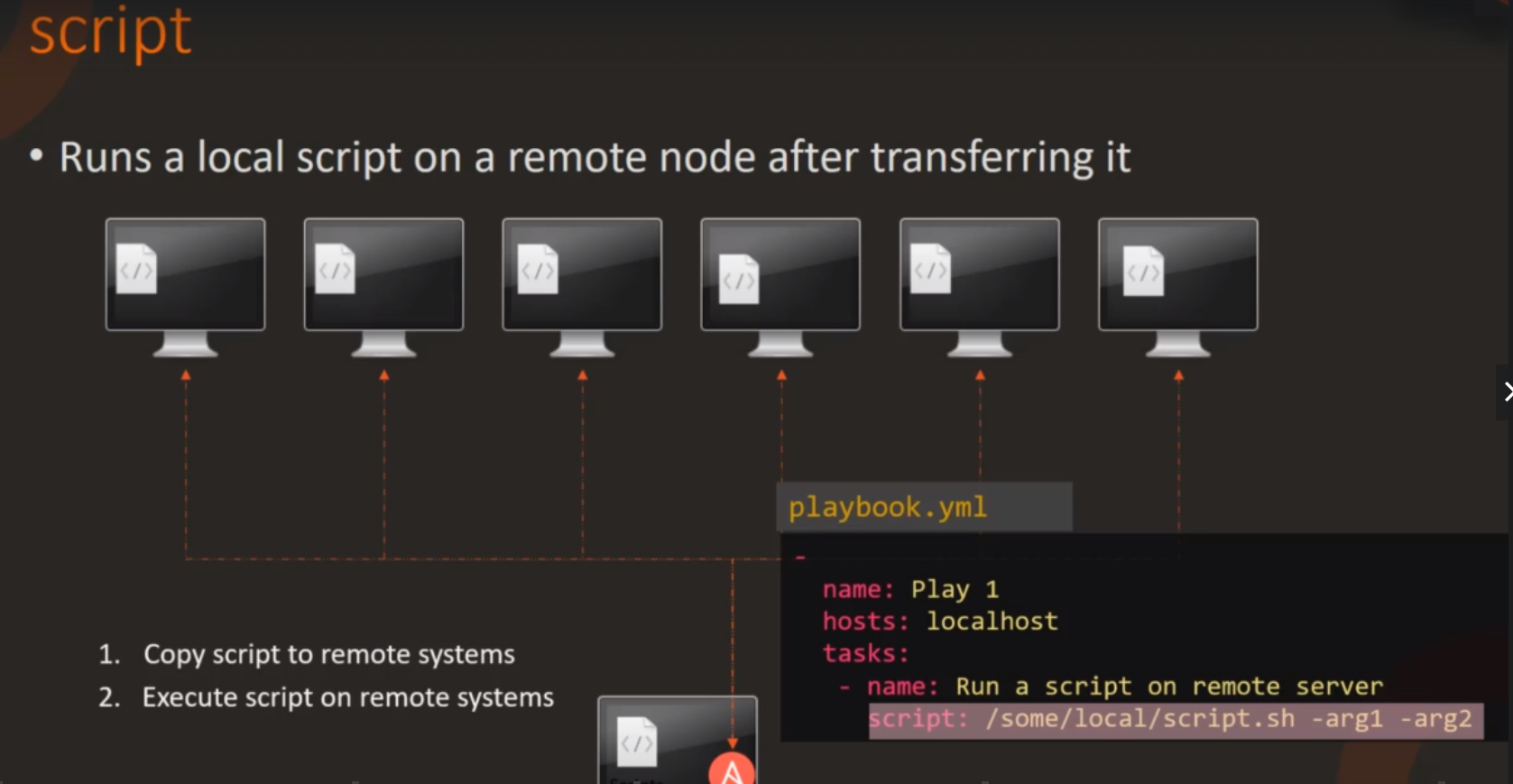
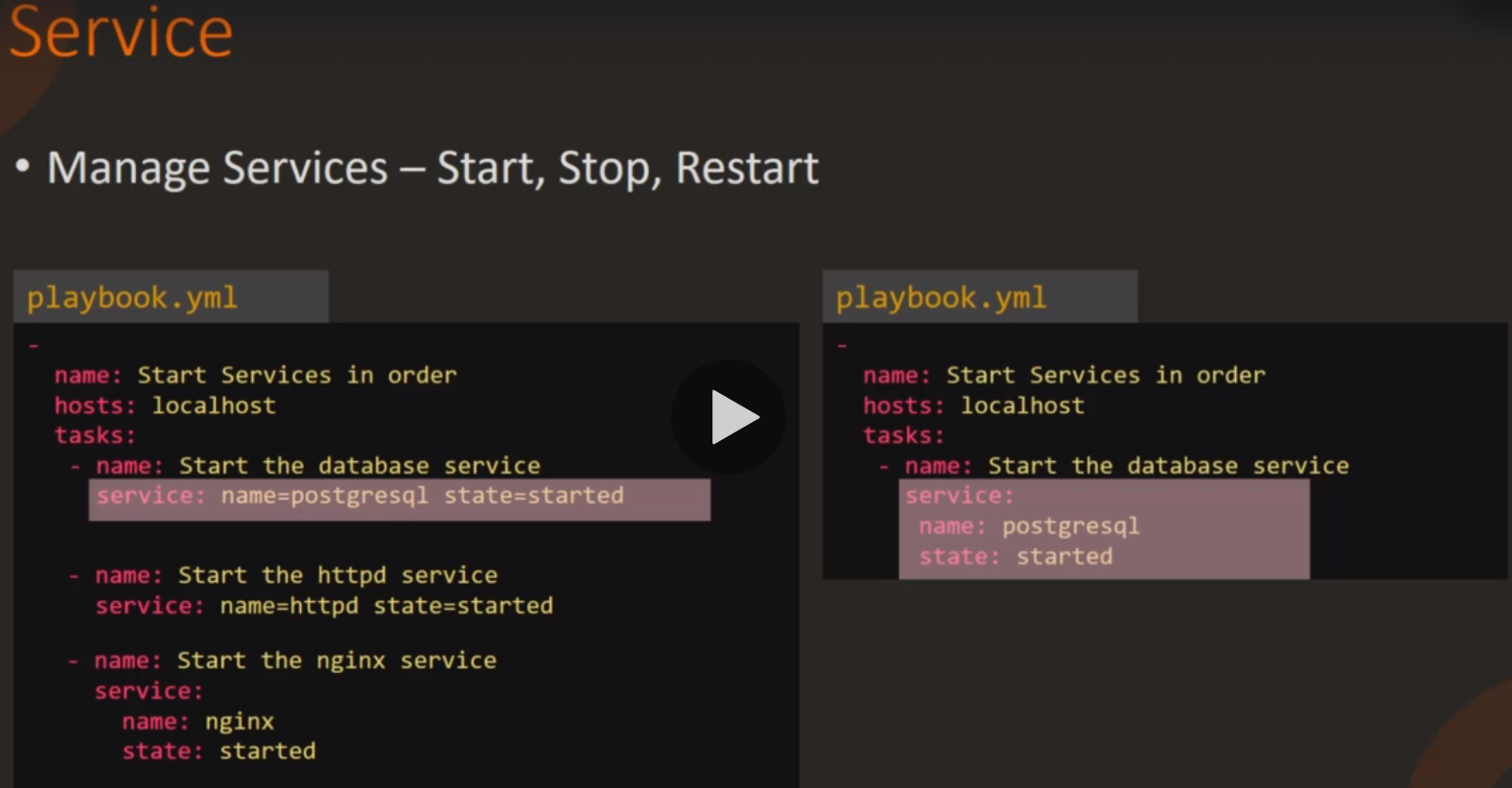
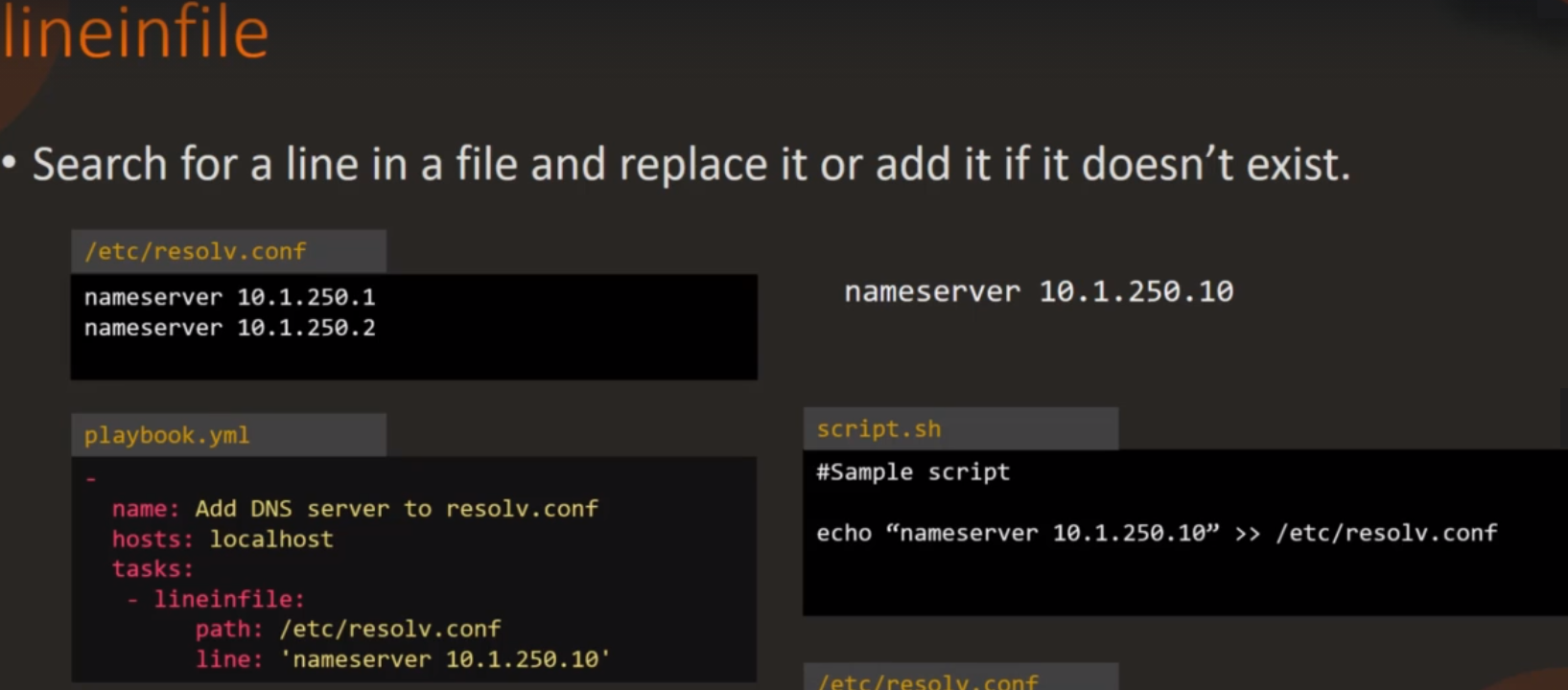
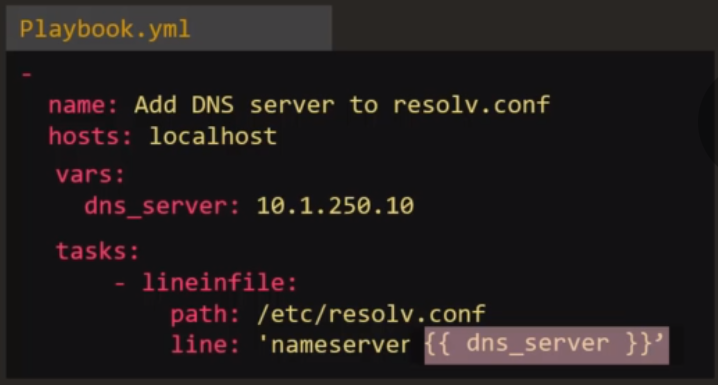
* **Inventory**
  + You only need to install Ansible on master (controller) , no need to install it on tagets
  + Ansible control machine can only be Linux and not windows (target m/c can be anything)
  + In Linux system Ansible talks to targets via SSH and in windows system through Powershell remoting
  + Information regarding this targets are stored into Inventory file.
  + Default location of inventory file is /etc/ansible/hosts  
      
      
* **Yaml** : yet another markup language

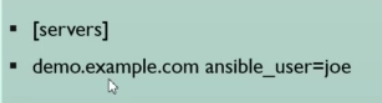
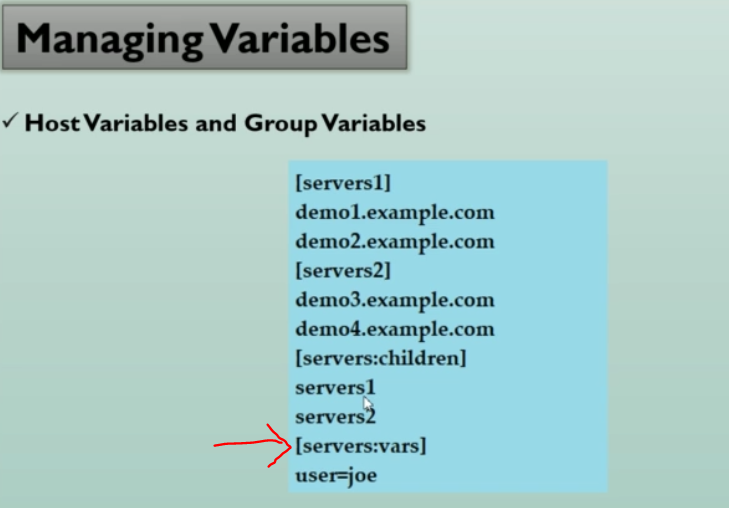
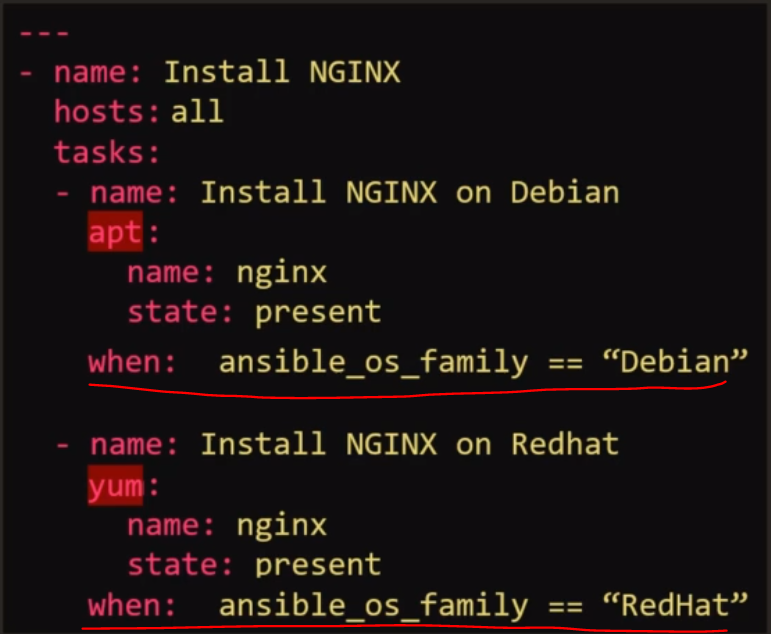
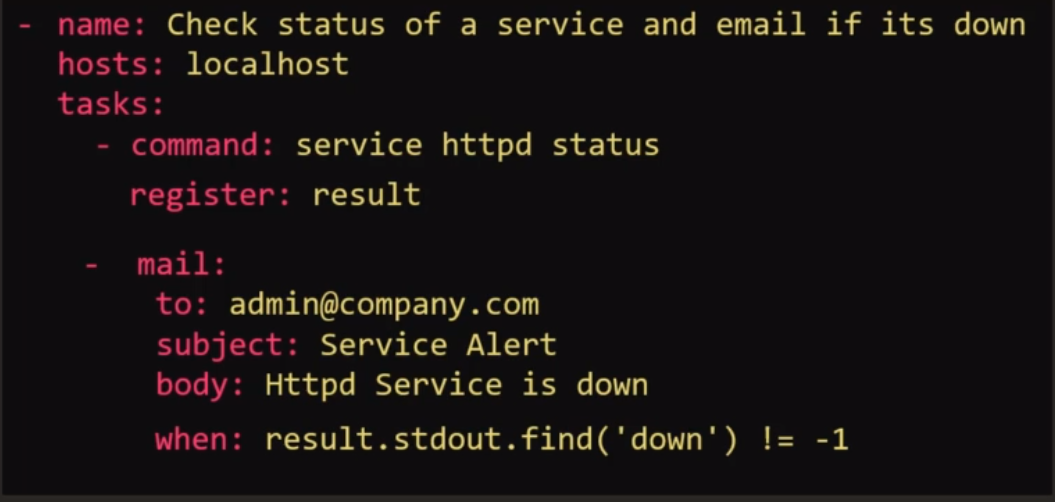
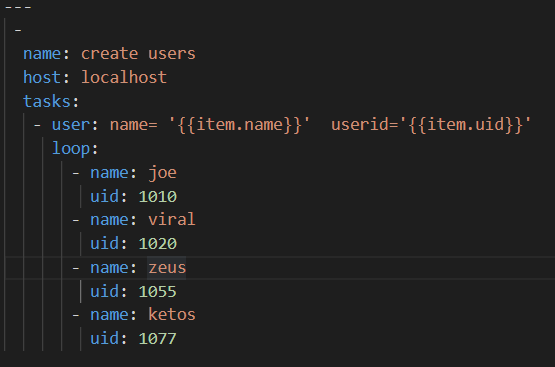
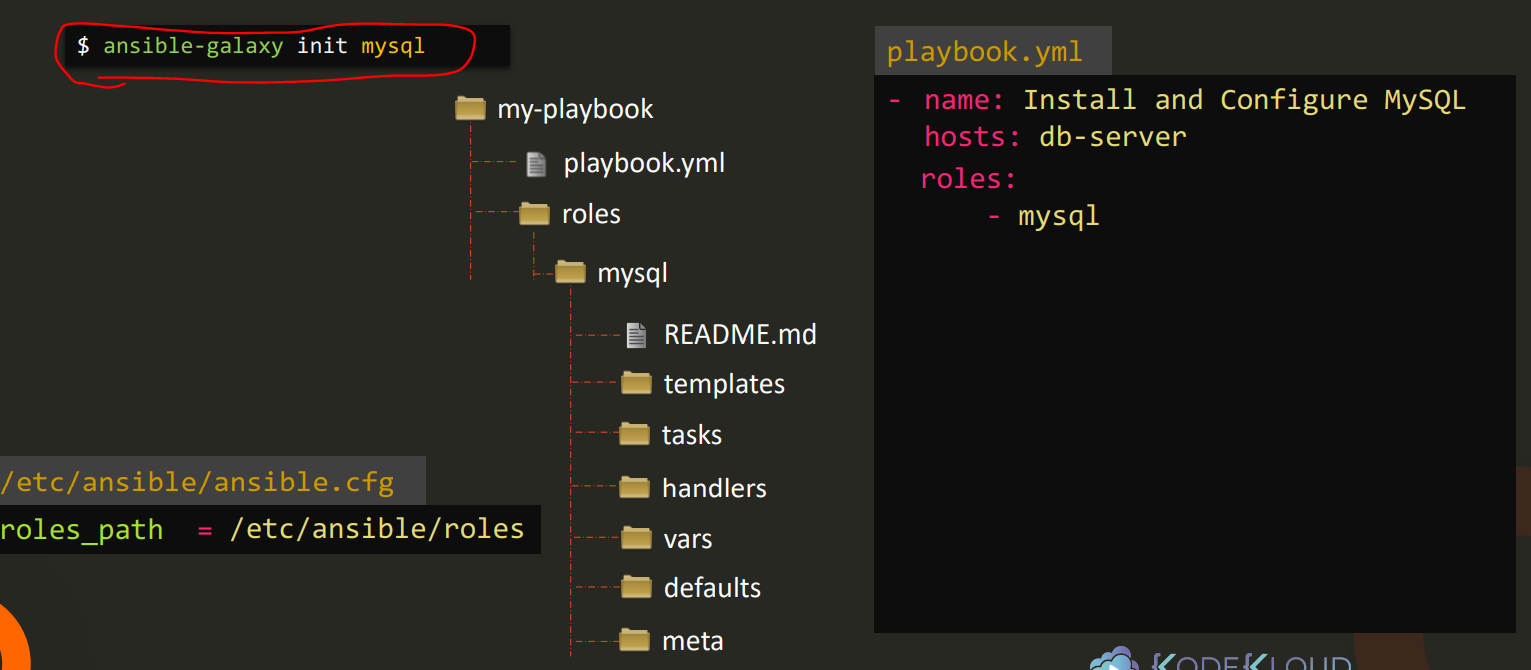


* 
* $ ansible-playbook playbook.yaml : to run the play book
* $ ansible target1 -m ping -i inventory.txt : to check connectivity with target1  
  
* **Modules**
  + Command module:   
    
  + Script module:

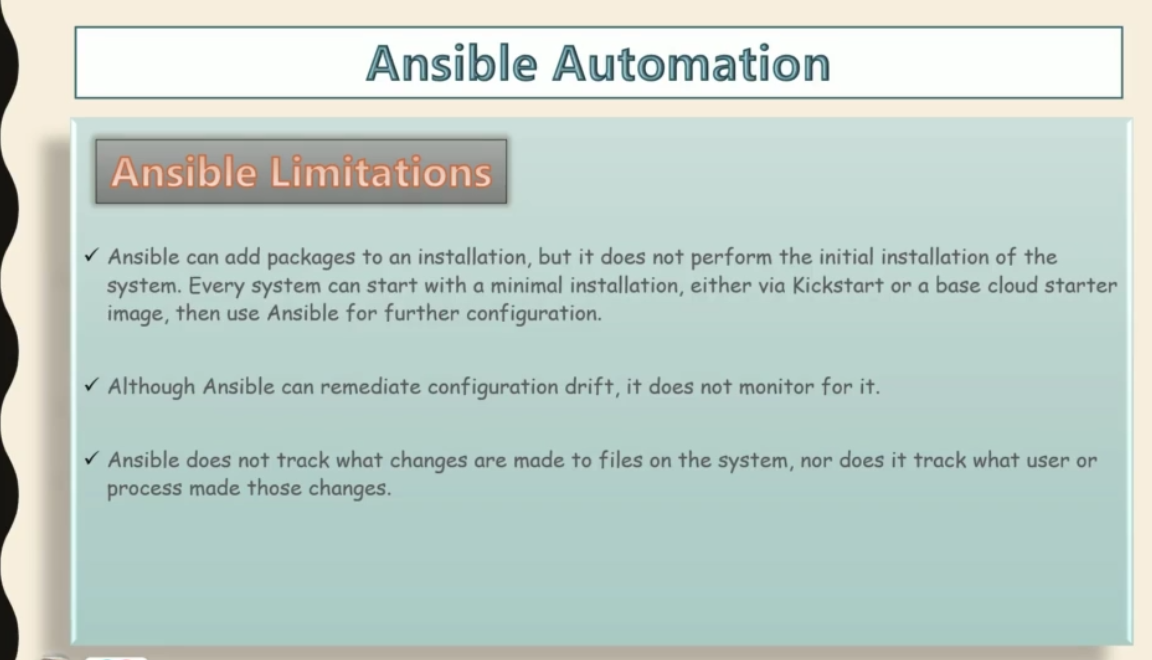


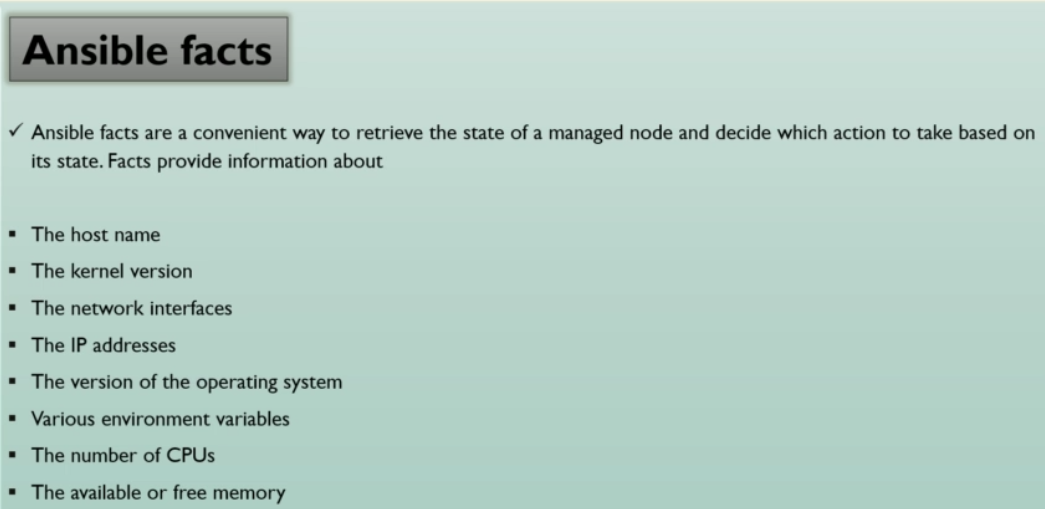
* + Service module:  
    
  + Lineinfile Module  
    
* **Variables:**



* You can also define vars in host or group in inventory file like  
   group variable (user) host variable (ansible\_user)
* **Conditional:**
  + 
  + 
* **Loops:**
  + When there is a repetitive values coming up , you can use Loop directive.
  + 
* **Roles:**
  + If you want to configure mysql server, then you will need to install pre-requisites, install Mysql packages, start Mysql service etc… so you can write down a play book containing all these task
  + But instead you can assign all those task under a role say ‘mysql’ and then simply use this role when ever you want to (so you won’t have to write down all the task evertime)
  + Ansible galaxy is portal in which you can find thousands of these kinds of roles.  
    
  + To create a role you need task, vars, defaults, handelrs, templates files.
  + Running $ ansible-galaxy init mysql will create the layout similar to this ,  
    

**Limitations of Ansible:**



**Ansible Facts:**  


**Ansible handler:**   
it is exactly same as tasks, but it will run when called by another task.  
ie. If you are writing a playbook which requires turning ON and OFF specific services again and again everytime, then rather than writing same task for turning ON/OFF services again and again, you can simply make a Handler and call it whenever require like this ,  
