

Κεφάλαιο 1 – RH294 - Ansible

Συμπληρωματικές σημειώσεις

Γεώργιος Μαγκλάρας PhD

Associate Instructor

RedHat Academy

Πανεπιστήμιο Δυτικής Αττικής



Ansible: Βασικές ιδέες

- Ο Κώστας έχει να διαχειριστεί 7 μηχανήματα: 2 file servers, 1 RDBMS και 4 Linux clients
- Ο Γιώργος έχει να διαχειριστεί 450 μηχανήματα: 20 file servers, 30 Application servers, 200 μηχανήματα Linux και άλλα 200 μηχανήματα MS-Windows
- Διαχείριση σημαίνει:
 - (Επαν)εγκατάσταση λειτουργικού
 - (Επαν)εγκατάσταση των απαραίτητων εφαρμογών/πακέτων, κτλ.
 - Παραμετροποίηση για λειτουργία στην υποδομή (δίκτυο, storage, printers)
 - Περιοδικές αναβαθμίσεις λειτουργικού και εφαρμογών πακέτων.
 - Monitoring για ασφάλεια και βέλτιστη απόδοση
 - Εκτέλεση εντολών σε όλα ή ορισμένα απο τα μηχανήματα
- Ο Κώστας έχει πάρει τις γνώσεις του RH124/134 και τα κάνει όλα με το χέρι.
- Ο Γιώργος έχει και αυτός τις ίδιες γνώσεις, δε μπορεί όμως να τα κάνει όλα με το χέρι.
- Η κλίμακα υποδομής παίζει βασικό ρόλο στην επιβολή αυτοματοποίησης
- Ερώτηση: Θα είχε νόημα ο Κώστας να αυτοματοποιήσει διαχειριστικά το περιβάλλον του δεδομένου του ότι η κλίμακα που δουλεύει είναι μικρή;
- Η αυτοματοποίηση απαιτεί πρόσθετη υποδομή και πρόσθετη γνώση. Είναι πάντοτε πιο αξιόπιστη λύση απο το “τα κάνω όλα με το χέρι, γιατί έχω 7 κουτάκια και που να κάθομαι εγώ τώρα να αυτοματοποιώ” για μικρές κλίμακες. Σε μεγάλες κλίμακες υποδομής, η αυτοματοποίηση είναι απαραίτητη.

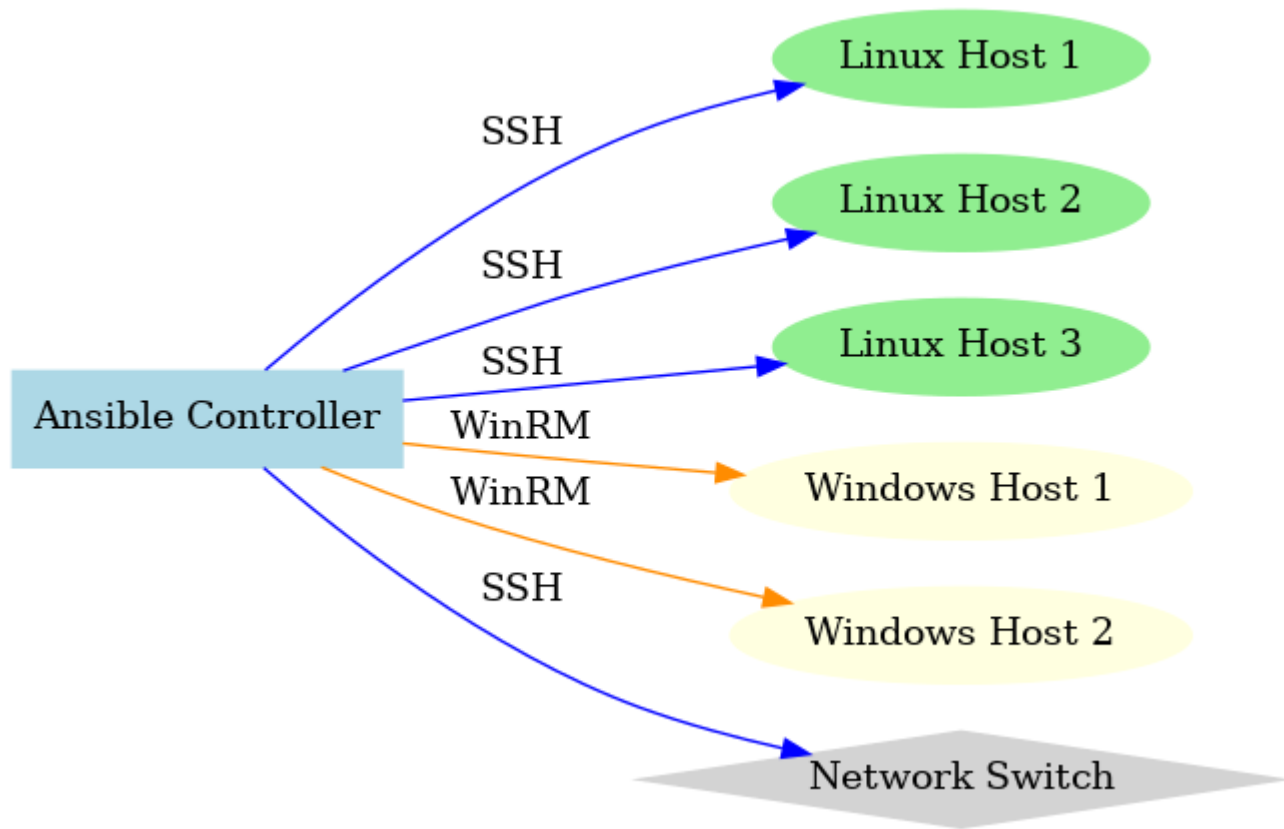


Ansible: Βασικές ιδέες (2)

- Το Ansible είναι μια γλώσσα αυτοματοποίησης υποδομών Πληροφορικής
- Είναι λογισμικό ανοικτού κώδικα κατα βάση, έστω και αν η RedHat προσφέρει προϊόντα που βασίζονται στο Ansible αλλά τα πακετάρει διαφορετικά για λόγους υποστήριξης και συμβατότητας.
- Ακολουθεί τη φιλοσοφία του **Infrastructure as Code (IaC)**: Ο διαχειριστικός κώδικας (λογικό) ως μέρος της υποδομής, πρέπει να αυτοματοποιεί αξιόπιστα και να είναι κατανοητός από άτομα που έχουν διαχειριστική γνώση. Η αξιοπιστία σημαίνει ότι
 - Η αυτοματοποίηση πρέπει να αφήνει συγκεκριμένα μέρη της υποδομής σε καλώς ορισμένη και αναπαραγώγιμη κατάσταση (well defined reproducible state)
 - Η αυτοματοποίηση πρέπει να έχει την ίδια αξιοπιστία σε μικρή και μεγάλη κλίμακα.
 - Οι παράμετροι και ο τρόπος που επιδρούν με την υποδομή πρέπει να είναι αγαγνώσιμοι/προσβάσιμοι από άτομα σχετικά με το αντικείμενο και να μην αφήνουν κενά. Για παράδειγμα, αν αύριο το πρωί ο εργαζόμενος διαχειριστής/ια φύγει από τη θέση του/της, ο καινούργιος/α να μπορεί να αναλάβει καθήκοντα με σχετική ευκολία.
- Το Ansible δεν είναι το μόνο εργαλείο IaC. Υπάρχουν και άλλα παρόμοια εργαλεία τα οποία δεν επιτυγχάνουν πάντοτε σε όλη τη διαχειριστική γκάμα διεργασιών και την απλότητα του Ansible, λ.χ.
 - [Puppet](#)
 - [Terraform](#)
 - [Progress Chef \(Chef\)](#)
- Το εύρος των διαχειριστικών εργασιών και η απλότητα του Ansible είναι τα δύο χαρακτηριστικά που το κάνουν ιδιαίτερα δημοφιλές και προτιμητέο για αυτοματοποίηση.

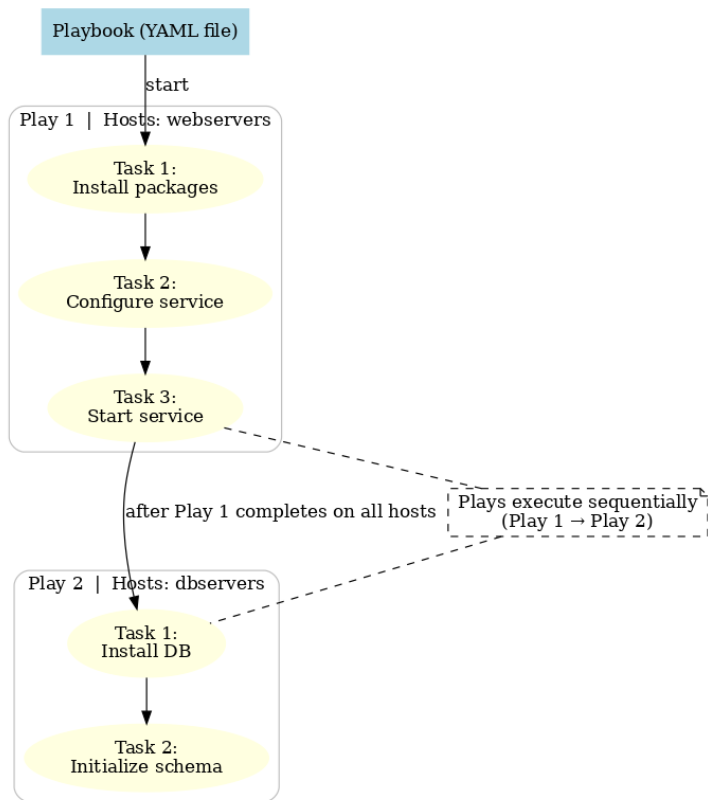


Ansible: Βασικές ιδέες (3)



- Agentless: Για να διαχειριστούμε συστήματα (managed systems) με το Ansible, δε χρειαζόμαστε να εγκαταστήσουμε πρόσθετο λογισμικό.
- Το Ansible πρέπει να εγκατασταθεί στον ελεγκτή (Ansible Controller).
- Εάν τα περισσότερα συστήματα έχουν Python και SSH, (αντίστοιχα WinRM για MS-Windows), αυτό αρκεί.
- Ο ελεγκτής μπορεί να είναι Linux ή MacOS, όχι όμως MS-Windows.
- Για συμβατότητα εκδόσεων Python στα managed hosts: https://docs.ansible.com/ansible/latest/reference_appendices/release_and_maintenance.html#ansible-core-support-matrix

Ansible: Βασικές ιδέες (4)



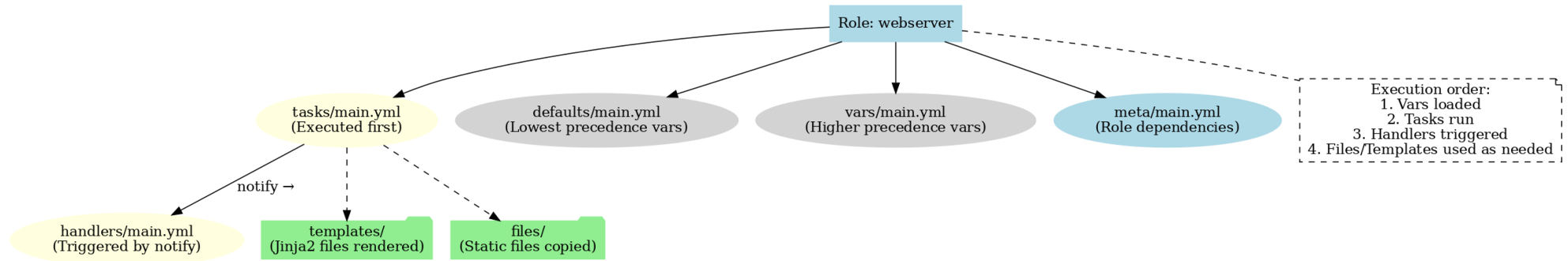
- Βασική ιδέα του Ansible είναι το αρχείο **playbook**.
- Περιέχει ένα ή περισσότερα **Plays**.
- Κάθε Play περιέχει μια ή περισσότερες εργασίες (**tasks**), οι οποίες αφορούν μια συγκεκριμένη ομάδα μηχανημάτων.
- Ένα **Ansible role** εκτελεί τις διαδικασίες πιο σύνθετα και με παραμέτρους (βλέπε slides 7 και 8)
- Οι ομάδες μηχανημάτων και το σύνολο αυτών καθορίζονται από ένα αρχείο απογραφής (**inventory file**)
- Η κάθε εργασία (task) εκτελεί κάτι συγκεκριμένο (εγκατάσταση ενός πακέτου/βάσης δεδομένων, κτλ) χρησιμοποιώντας ένα συγκεκριμένο **Ansible module**.
- Οι εργασίες σε κάθε Play εκτελούνται σειριακά, με την σειρά που είναι γραμμένες. Το ίδιο ισχύει και για τα Plays.
- Η σύνταξη του κάθε Playbook ακολουθεί τους κανόνες της **YAML**.

Ansible: Παράδειγμα playbook

```
and its role is to set authentication for the hosts.
[georgios@cn1 hpcansible]$ cat setauthentication.yml
- name: Setup users and groups
  hosts:
    - cn1
    - cn2
    - mds
  become: true

  tasks:
    - name: Copy/sync the /etc/passwd with owner and permissions
      ansible.builtin.copy:
        src: files/authentication/passwd
        dest: /etc/toconcatpasswd
        owner: root
        group: root
        mode: '0644'
    - name: Copy/sync the /etc/group with owner and permissions
      ansible.builtin.copy:
        src: files/authentication/group
        dest: /etc/toconcatgroup
        owner: root
        group: root
        mode: '0644'
    - name: Concatenate to the /etc/passwd if the info does not exist
      ansible.builtin.shell: for i in $(cat /etc/toconcatpasswd); do grep -qF -- "$i" /etc/passwd || echo "$i" >> /etc/passwd; done
    - name: Concatenate to the /etc/group if the info does not exist
      ansible.builtin.shell: for i in $(cat /etc/toconcatgroup); do grep -qF -- "$i" /etc/group || echo "$i" >> /etc/group; done
[georgios@cn1 hpcansible]$
```

Ansible: Βασικές ιδέες (5)



- Σε πλήρη αντίθεση με τη σειριακή συσχέτιση εργασιών με ομάδες μηχανημάτων στα βασικά playbooks, υπάρχει και ο μηχανισμός του **Ansible role**.
- Είναι μια αυτόνομη, φορητή μονάδα αυτοματισμού που ομαδοποιεί σχετικές εργασίες και τις συσχετίζει με μεταβλητές, αρχεία, χειριστές (handlers) σε μια καλώς ορισμένη δομή.
- Αυτή η δομή του Ansible role, επιτρέπει την επαναχρησιμοποίηση κώδικα (code reusability)
- Η χρήση του Ansible role προσδίδει ευελιξία και επιτρέπει την επίτευξη πολύπλοκων κανόνων αυτοματισμού (π.χ. Αν το λειτουργικό είναι RHEL 9, τότε εγκατέστησε αυτά τα αρχεία και πακέτα, εάν είναι RHEL 10, εγκατέστησε τα άλλα αρχεία και πακέτα)
- <https://www.redhat.com/en/topics/automation/what-is-an-ansible-role>

Ansible: Παράδειγμα ansible role (5)

```
# site.yml
- hosts: webserver
  tasks:
    - name: Install nginx
      yum:
        name: nginx
        state: present
```

Απλό playbook με ένα task

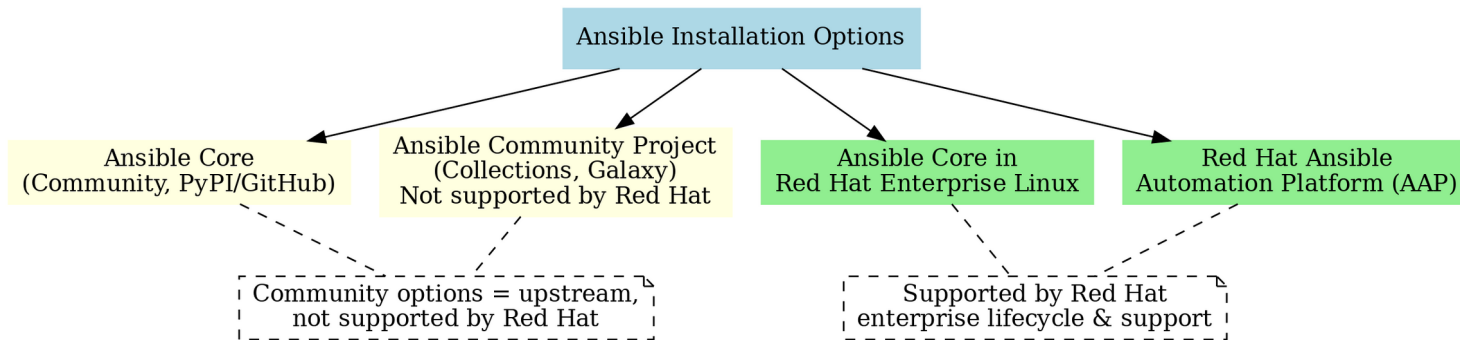
```
# site.yml
- hosts: webserver
  roles:
    - webserver
```

Ansible role ▲ με τη δομή αρχείων ▼

```
roles/
  webserver/
    tasks/
      main.yml # same "install nginx" task lives here
    handlers/
    vars/
    defaults/
    templates/
    files/
    meta/
```

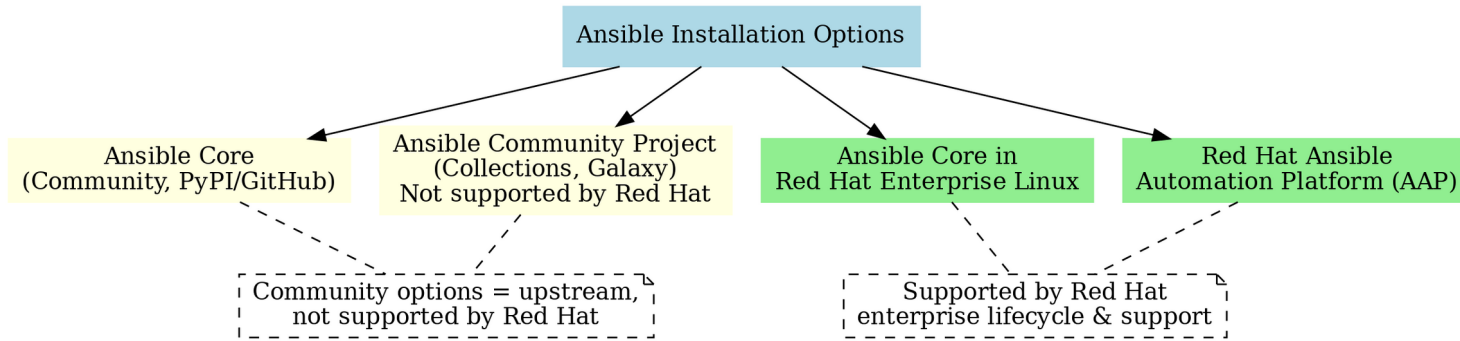
- Εάν πρέπει να εγκαταστήσουμε τον nginx σε μια ομοιογενή ομάδα μηχανών (webserver → όλα RHEL 9), τότε ένα απλό playbook με ένα task αρκεί.
- Εάν όμως πρέπει να πράξουμε το ίδιο σε μια ανομοιογενή ομάδα μηχανών (webserver → 5xRHEL 9, 2xRHEL10, 3xUbuntu Noble), τότε είναι προτιμότερο να κάνουμε χρήση ενός ρόλου και της δομής αρχείων του, η οποία θα καθορίζει σε κάθε περίπτωση το πως θα γίνει η εγκατάσταση σε διαφορετικά λειτουργικά.

Ansible: Εκδόσεις (versions)



- **Ansible core:** Το περιβάλλον εκτέλεσης του Ansible μαζί με έναν μικρό αριθμό απο modules για βασικές εργασίες (ansible builtin). Δεν υποστηρίζεται απο τη Redhat αλλά απο την κοινότητα ανοικτού λογισμικού.
- **Ansible Community Project:** Το ansible core μαζί με πληθώρα modules (Collections) επιλεγμένων απο την κοινότητα ανοικτού λογισμικού. Δεν υποστηρίζεται απο τη Redhat αλλά απο την κοινότητα ανοικτού λογισμικού.
- **Ansible core in Red Hat Enterprise Linux:** Το RPM πακέτο του περιβάλλοντος εκτέλεσης του Ansible, με modules προσαρμοσμένα στην εκτέλεση playbooks για προϊόντα Redhat. Σε αυτά σταματάει η υποστήριξη της Redhat.
- **Ansible Automation Platform*:** Πλήρης υποστήριξη του Ansible Core καθώς και επιπλέον πιστοποιημένο και υποστηριζόμενο περιεχόμενο, εργαλεία, στοιχεία και υπηρεσίες cloud.

Ansible: Εκδόσεις (versions) (2)



- Στις ασκήσεις μας στα εργαστήρια/Virtual Machines του μαθήματος RH294, θα εξετάζουμε τις εκδόσεις AAP και Ansible Core in Red Hat Enterprise Linux.
- Πέραν αυτών των επίσημων ασκήσεων, κατά τη διάρκεια των διαλέξεων θα δίνουμε και παραδείγματα με τα Ansible Core/Ansible Community Project που μπορείτε να χρησιμοποιείτε και σε διανομές πέραν του RedHat Enterprise Linux
- Ο τρόπος/σύνταξη που γράφουμε όλο των κώδικα σε Ansible είναι ο ίδιος σε όλες τις εκδόσεις του. Άρα ότι μάθουμε για το AAP ισχύει και για τις community εκδόσεις.
- Η ουσιαστική διαφορά είναι ο τρόπος που τρέχουμε τον κώδικα:
 - Για την μεν RedHat AAP και το Ansible Core in RedHat Enterprise Linux υπάρχει η εντολή **ansible-navigator run** η οποία τρέχει το περιβάλλον απο container images
 - Για τη δε community version χρησιμοποιείται η εντολή **ansible-playbook** που δεν χρησιμοποιεί container images

Ansible roles – Δομή φακέλων

```
(hpcansible) [georgios@cn1 lighttpd_gcc]$ tree -F
.
├── defaults/
│   └── main.yml
├── handlers/
│   └── main.yml
├── meta/
│   └── main.yml
├── tasks/
│   ├── debian.yml
│   ├── main.yml
│   └── rhel.yml
└── templates/
    └── lighttpd.conf.j2

5 directories, 7 files
```

- **Defaults/main.yml** → Ορισμός μεταβλητών με εμβέλεια (scope) ρόλου
- **handlers/main.yml** → Εργασίες χειρισμού (handlers), οι οποίες συσχετίζουν την εκτέλεση μιας εργασίας (task) με την εκτέλεση μιας άλλης. Πχ: Αν μεταβληθεί το αρχείο παραμέτρων μιας υπηρεσίας (service), επανεκκίνησε την υπηρεσία
- **meta/main.yml** → Πληροφορίες για το ρόλο (συγγραφέας, περιγραφή του τι πράττει ο ρόλος, απαιτούμενες εκδόσεις, κτλ)
- **tasks/main.yml** → Σύνολο των εργασιών που εκτελεί ο ρόλος συνήθως κατηγοριοποιημένες ανάλογα με την πλατφόρμα/διανομή του λειτουργικού συστήματος
- **Templates/** → Πρότυπα λογισμικού που επιτρέπουν τη δημιουργία αρχείων παραμετροποίησης σχετικών με τις εργασίες που εκτελεί ο ρόλος

Ansible roles – Παράδειγμα

Κατασκευή ενός ρόλου Ansible με τις ακόλουθες εργασίες/ιδιότητες

- Εγκατάσταση και παραμετροποίηση του διακομιστή ιστοσελίδων lighttpd
- Εγκατάσταση του μεταγλωττιστή gcc

σε λειτουργικά συστήματα RedHat Enterprise Linux και Ubuntu Linux, ακολουθώντας τη δομή αρχείων του slide 10.

Το παράδειγμα είναι διαθέσιμο [εδώ](#) .

Ansible roles – Παράδειγμα (2)

```
1 # You can override these in your play/group/host vars as needed
2 lighttpd_config_path: /etc/lighttpd/lighttpd.conf
3 lighttpd_service_name: lighttpd
4 lighttpd_packages:
5   - lighttpd
6   - gcc
7
8 # Example minimal config options for the template
9 lighttpd_server_port: 80
10 lighttpd_server_root: /var/www/html
11 lighttpd_index_files: ["index.html"]
12
```

- Στο `Defaults/main.yml` αρχείο καθορίζουμε μεταβλητές εμβέλειας ρόλου όπως
 - `lighttpd_config_path`: Που θα τοποθετηθεί το αρχείο παραμετροποίησης του lighttpd
 - `lighttpd_service_name`: Το όνομα της υπηρεσίας/διακομιστή ιστοσελίδων
 - `lighttpd_packages`: μεταβλητή τύπου λεξικού (dictionary) που ορίζει ποια πακέτα θα εγκατασταθούν
 - `lighttpd_server_port`: Ποια πόρτα (tcp) θα χρησιμοποιήσει ο διακομιστής ιστοσελίδων
 - `lighttpd_server_root`: Σε ποιο φάκελο θα βρίσκει τα αρχεία που πρέπει να δίνει ο διακομιστής ιστοσελίδων
 - `lighttpd_index_files`: Ποιο θα είναι το αρχείο δείκτης (index), το πρώτο αρχείο που δίνεται ερήμην στο χρήστη στο φάκελο απο το διακομιστή ιστοσελίδων

Ansible roles – Παράδειγμα (3)

```
1 ---
2 - name: restart lighttpd
3   ansible.builtin.service:
4     name: "{{ lighttpd_service_name }}"
5     state: restarted
```

- Στο `handlers/main.yml` αρχείο έχουμε μια και μόνη εργασία χειρισμού
 - Όταν καλεστεί με το **notify: restart lighttpd** θα εκτελεστεί και θα επανεκκινήσει την υπηρεσία `lighttpd.service` (βλέπε `tasks/main.yml`)
 - Χρησιμοποιεί το [ansible.builtin.service module](#)
 - Η μεταβλητή `lighttpd_service_name` καθορίζει το όνομα της υπηρεσίας που θα επανεκκινηθεί (`lighttpd`). Ο λόγος που χρησιμοποιούμε μεταβλητή εδώ γίνεται κατανοητός εαν κάνουμε τις ακόλουθες ερωτήσεις:
 - Εάν άλλαζε το όνομα του πακέτου, θα χρειαζόμασταν να αλλάξουμε κάτι στην εργασία χειρισμού;

Ansible roles – Παράδειγμα (4)

```
1 ---
2 galaxy_info:
3   role_name: lighttpd_gcc
4   author: Georgios Magklaras
5   description: Install lighttpd and gcc on Ubuntu/RHEL8, deploy config, and restart on change.
6   license: MIT
7   version: 1.0
8   min_ansible_version: "2.12"
9   platforms:
10    - name: EL
11      versions: ["8"]
12    - name: Ubuntu
13      versions: ["focal", "jammy", "noble"]
14 dependencies: []
15
```

- Στο `meta/main.yml` αρχείο έχουμε όλες τις πληροφορίες για τον ρόλο
 - Οι πληροφορίες αυτές είναι χρήσιμες για τους χρήστες του ρόλου
 - Διαβάζονται και χρησιμοποιούνται απο εργαλεία όπως το [ansible-galaxy](#)
 - Οι ρόλοι συμβάλλουν στη δημιουργία αρθρωτού κώδικα και στην επαναχρησιμοποίησή του (modularity, code reusability). Αυτές οι πληροφορίες επιτρέπουν στη σωστή κατηγοριοποίησή του ρόλου, ώστε να μπορούν να τον εντοπίσουν οι χρήστες απο `ansible collections` και λοιπά `software repositories`.

Ansible roles – Παράδειγμα (5)

```
georgios@cn1:~/lustre-hpc-test-setup/hpcansible/roles/lighttpd_gcc/tasks/78x22
1 --
2 # RHEL8/derivatives specific: ensure EPEL is present for lighttpd
3 - name: Ensure EPEL is enabled on RHEL 8
4   ansible.builtin.package:
5     name: epel-release
6     state: present
7   when:
8     - ansible_facts['os_family'] == 'RedHat'
9     - ansible_facts['distribution_major_version'] | int == 8
10
11 # Optional: enable CodeReady/CRB - useful for some dependencies
12 # This is best-effort and will not fail the run if unavailable.
13 - name: Enable CodeReady Builder on RHEL 8 (best-effort)
14   ansible.builtin.command: >
15     subscription-manager repos --enable codeready-builder-for-rhel-8-{{ a
16   changed_when: "'enabled' in stdout"
17   failed_when: false
18   when:
19     - ansible_facts['distribution'] == 'RedHat'
20     - ansible_facts['distribution_major_version'] | int == 8
```

```
georgios@cn1:~/lustre-hpc-test-setup/hpcansible/roles/lighttpd_gcc/tasks
georgios@cn1:~/lustre-hpc-test-setup/hpcansible/roles/lighttpd_gcc/tasks 78x22
```

```
1 --
2 # Debian/Ubuntu-specific preparation (optional apt cache update)
3 - name: Update apt cache (Debian/Ubuntu)
4   ansible.builtin.apt:
5     update_cache: yes
6     cache_valid_time: 3600
7
```

tasks/rhel.yml και tasks/debian.yml είναι δύο playbooks που θα συμπεριληφθούν στο tasks/main.yml. Εδώ βλέπουμε πως ο ρόλος μας δίνει τη δυνατότητα να κάνουμε αρθρωτό κώδικα. Άλλες εργασίες χρειάζονται ως προεργασία για τη διανομή RHEL και άλλες για την Ubuntu.

Ansible roles – Παράδειγμα (6)

```
georgios@trichonida:~  
georgios@cn1:~/lustre-hpc-test-setup/hpcansible  
1 ---  
2 - name: Include OS-family specific prep  
3   ansible.builtin.include_tasks: debian.yml  
4   when: ansible_facts['os_family'] == 'Debian'  
5  
6 - name: Include RHEL-specific prep  
7   ansible.builtin.include_tasks: rhel.yml  
8   when:  
9     - ansible_facts['os_family'] == 'RedHat'  
10    - ansible_facts['distribution_major_version'] | int == 8  
11  
12 - name: Install packages (portable across distros)  
13   ansible.builtin.package:  
14     name: "{{ httpd_packages }}"  
15     state: present  
16  
17 - name: Deploy httpd config  
18   ansible.builtin.template:  
19     src: httpd.conf.j2  
20     dest: "{{ httpd_config_path }}"  
21     owner: root  
22     group: root  
23     mode: "0644"  
24     backup: yes  
25   notify: restart httpd  
26  
27 - name: Ensure httpd is enabled and running  
28   ansible.builtin.service:  
29     name: "{{ httpd_service_name }}"  
30     state: started  
31     enabled: yes
```

• tasks/main.yml

- Ο συνδετικός κρίκος που περιλαμβάνει τις (προ)εργασίες των tasks/rhel.yml και tasks/debian.yml (βλέπε προηγούμενο slide)
- Το `ansible.builtin.package` είναι distribution agnostic. Απαραίτητη επιλογή αφού ο ρόλος διαχειρίζεται εργασίες και σε RHEL και σε Ubuntu targets.
- Το `ansible.builtin.template` θα δημιουργήσει δυναμικά το αρχείο παραμετροποίησης του httpd (/etc/httpd/httpd.conf, βλέπε επόμενο slide). Θα καλέσει επίσης την εργασία χειρισμού (handler) του handlers/main.yml για να επανεκκινήσει το διακομιστή ιστοσελίδων κάθε φορά που αλλάζει το αρχείο
- Ερώτηση: Τι ακριβώς κάνει το `ansible.builtin.service` εδώ σε αυτό το παράδειγμα;

Ansible roles – Παράδειγμα (7)

```
georgios@trichonida:~  
georgios@cn1:~/lustre-hpc-test-setup/hpcansible/roles/lighttpd_gcc/templates 87x37  
# Managed by Ansible – do not edit manually.  
server.port = {{ lighttpd_server_port }}  
server.document-root = "{{ lighttpd_server_root }}"  
index-file.names = [ {% for f in lighttpd_index_files %}"{{ f }}"{% if not loop.last %}  
, {% endif %}{% endfor %} ]  
  
# Minimal sane defaults  
server.username = "www-data"  
server.groupname = "www-data"  
dir-listing.activate = "disable"  
  
# Log locations (defaults)  
accesslog.filename = "/var/log/lighttpd/access.log"  
server.errorlog = "/var/log/lighttpd/error.log"  
  
# Include additional conf.d if present  
include_shell "/usr/share/lighttpd/create-mime.conf.pl"  
include "/etc/lighttpd/conf-enabled/*.conf"
```

- **templates/lighttpd.conf.j2**

- Ο ρόλος ενός προτύπου Ansible είναι η δυναμική δημιουργία αρχείων παραμετροποίησης σε μηχανήματα στόχους χρησιμοποιώντας μια γλώσσα προγραμματισμού που βασίζεται στη συντακτική φόρμα [Jinja2](#).
- Η δυναμική δημιουργία του περιεχομένου του αρχείου επιτυγχάνεται με την αντικατάσταση των τιμών των μεταβλητών.
- Στην ουσία, λέμε:
 - Πάρε τη δομή του αρχείου
 - Αντικατέστησε την τιμή των μεταβλητών
 - Βάλε το έτοιμο πλέον αρχείο εκεί που ορίζει το `ansible.builtin.template module` (βλέπε προηγούμενο slide)
- Το templating (πρότυπο) είναι τεχνική που χρησιμοποιείται και σε άλλα συστήματα λογισμικού εκτός του Ansible, προσθέτοντας ευελιξία στη δημιουργία δεδομένων/αρχείων.

Ansible roles – Παράδειγμα (8)

georgios@cn1:~/lustre-hpc-test-setup/hpcansible 64x36

```
1 - name: Install and configure lighttpd + gcc
2   hosts: athens
3   become: yes
4   gather_facts: yes
5
6   roles:
7     - role: lighttpd_gcc
8       vars:
9         # Optional overrides:
10        # lighttpd_server_port: 8080
11        # lighttpd_server_root: /srv/www
12        # lighttpd_index_files: ["index.html", "index.htm"]
```

georgios@cn1:~/lustre-hpc-test-setup/hpcansible 92x36

```
(hpcansible) [georgios@cn1 hpcansible]$ ansible-playbook -i inventory/ role1.yml

PLAY [Install and configure lighttpd + gcc] *****

TASK [Gathering Facts] *****
ok: [cn1]
ok: [cn2]
ok: [cn5]

TASK [lighttpd_gcc : Include OS-family specific prep] *****
skipping: [cn1]
skipping: [cn2]
included: /home/georgios/lustre-hpc-test-setup/hpcansible/roles/lighttpd_gcc/tasks/debian.yml for cn5

TASK [lighttpd_gcc : Update apt cache (Debian/Ubuntu)] *****
changed: [cn5]

TASK [lighttpd_gcc : Include RHEL-specific prep] *****
skipping: [cn5]
included: /home/georgios/lustre-hpc-test-setup/hpcansible/roles/lighttpd_gcc/tasks/rhel.yml for cn1, cn2

TASK [lighttpd_gcc : Ensure EPEL is enabled on RHEL 8] *****
ok: [cn2]
ok: [cn1]
```

Εκτέλεση του ρόλου (δεξιά) απο το αρχείο role1.yml (αριστερά)

Ansible roles – Επιλογές εγκατάστασης

Έχουμε συνήθως δύο επιλογές για να εγκαταστήσουμε/χρησιμοποιήσουμε ρόλους

- A) Να τον γράψουμε με τη δομή φακέλων που έδειξε το παράδειγμα των προηγούμενων slides
- B) Να κατεβάσουμε το ρόλο από εξωτερικές πηγές, είτε:
 - ως μέρος των συλλογών (Ansible collections)
 - ως git repository
- Η φιλοσοφία των Ansible roles επιβάλλει να εγκαθιστούνται ως επαναχρησιμοποιήσιμα κομμάτια κώδικα, τα οποία άλλοι χρήστες μπορούν να αναζητήσουν, κατεβάσουν και εγκαταστήσουν προκειμένου να ικανοποιήσουν μια ανάγκη αυτοματοποίησης στα συστήματα. Κάποιος άλλος έχει μπει στον κόπο να γράψει το ρόλο λύνοντας το πρόβλημα.
- Πρωταρχικό ρόλο σε αυτές τις ανάγκες παίζει η εντολή [ansible-galaxy](#).

Ansible roles – Επιλογές εγκατάστασης (2)

- Παράδειγμα: Έχω ένα ρόλο και θέλω να τον κάνω διαθέσιμο σε άλλους χρήστες μέσω του github.
- Βήμα 1: Δημιουργώ τη δομή των φακέλων του ρόλου:

```
export ROLE_NAME=lighttpd_gcc
mkdir -p ansible-role-$ROLE_NAME
cd ansible-role-$ROLE_NAME
ansible-galaxy init lighttpd_gcc
```

- Βήμα 2: Γράφω/δημιουργώ το ρόλο όπως στα προηγούμενα slides
- Βήμα 3: Ανεβάζω το ρόλο στο github.com

```
git remote set-url origin https://github.com/gmagklaras/ansible-role-lighttpd_gcc.git
git commit -m "Initial commit"
git push -u origin main --force
```

- Βήμα 4: Για να τον χρησιμοποιήσουν οι χρήστες, τους συμβουλεύω να δημιουργήσουν το αρχείο requirements.yml:

```
- name: lighttpd_gcc
  src: https://github.com/gmagklaras/ansible-role-lighttpd_gcc.git
  version: main
```

- Βήμα 5: **ansible-galaxy role install -r requirements.yml -p roles**

Ansible roles – Επιλογές εγκατάστασης (3)

The screenshot shows the Ansible Galaxy website interface. On the left is a navigation menu with options: Ansible Galaxy, Search, Collections, Roles, Role Namespaces, Documentation, and Terms of Use. The 'Roles' section is currently selected. The main content area is titled 'Roles' and features a search bar with the keyword 'chronyd' entered. Below the search bar, there are two search results. The first result is 'autechgemz.chronyd', provided by 'autechgemz', with a description 'Manage chrony Network Time Protocol Service'. It has 'Updated 2 years ago' and '91 Downloads'. The second result is 'vdzhorov.ansible_role_chronyd', provided by 'vdzhorov', with a description 'Role for installing and configuring chronyd on CentOS 7 systems'. It has 'Updated 2 years ago' and '56 Downloads'. Below the description of the second role, there are three tags: 'chronyd', 'linux', and 'ntp'.

Το ίδιο πράγμα στο command line επιτυγχάνεται με: **ansible-galaxy search chronyd**
Για εγκατάσταση: **ansible-galaxy role install vdzhorov.ansible_role_chronyd**
Για απεγκατάσταση: **ansible-galaxy role remove vdzhorov.ansible_role_chronyd**

Ερωτήσεις

