



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

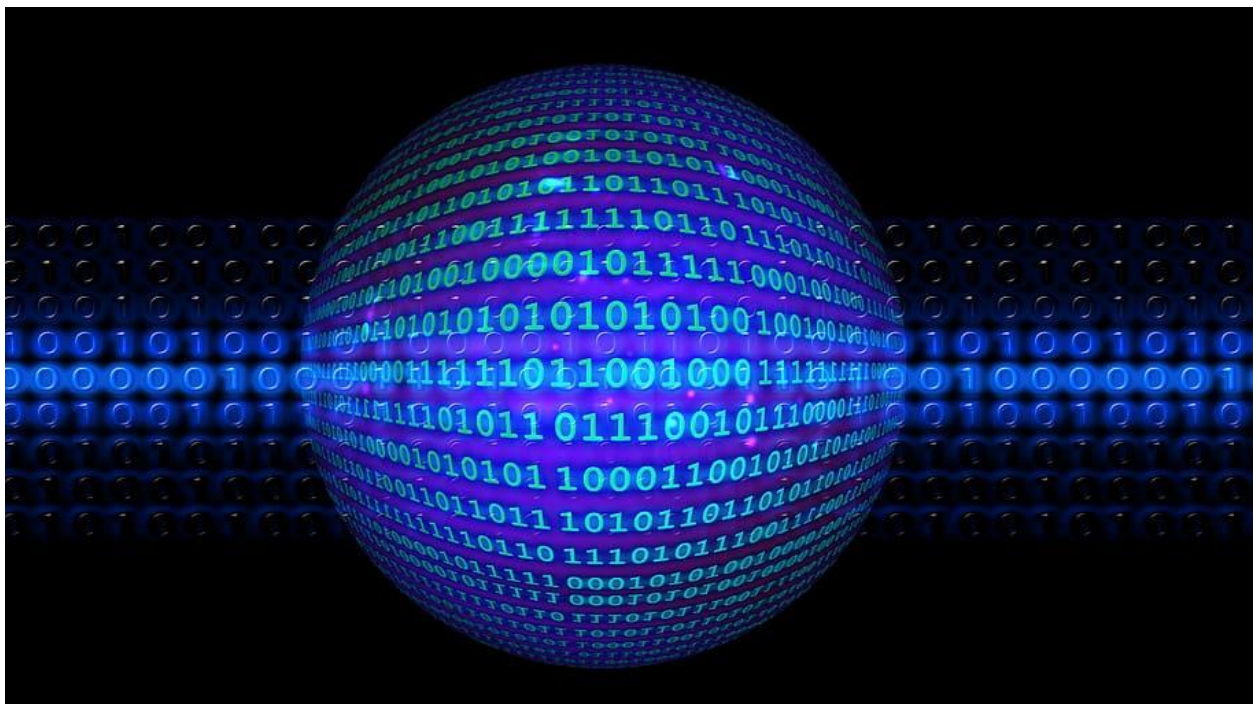
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗ &
ΤΗΛΕΜΑΤΙΚΗ

ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΣΤΗ ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ

**Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση
Ansible σε περιβάλλον MS Windows**

Διπλωματική Εργασία

Ελισάβετ Αραμπατζή



Αθήνα, Ιούλιος 2020



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΠΛΗΡΟΦΟΡΙΚΗ &
ΤΗΛΕΜΑΤΙΚΗ**

ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΣΤΗ ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ

Τριμελής Εξεταστική Επιτροπή

Βασίλειος Δαλάκας (Επιβλέπων)

Ε.ΔΙ.Π., Τμήμα Πληροφορικής & Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

Θωμάς Καμαλάκης

**Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής & Τηλεματικής, Χαροκόπειο
Πανεπιστήμιο**

Ανάργυρος Τσαδήμας

Ε.ΔΙ.Π., Τμήμα Πληροφορικής & Τηλεματικής, Χαροκόπειο Πανεπιστήμιο

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Η Ελισάβετ Αραμπατζή δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Στους γονείς μου, στον αδερφό μου και στο σύντροφό μου, οι οποίοι με στήριξαν με κάθε τρόπο και ήταν πάντα δίπλα μου και σε αυτό το ακαδημαϊκό μου ταξίδι...

“Στέργε μὲν τὰ παρόντα, ζήτει δέ τὰ βελτίω”.

Ισοκράτης (436 π.Χ. - 338 π.Χ.)

“Αυτά που έχεις επιτύχει να προσπαθείς να τα διατηρείς, αλλά συγχρόνως να αναζητάς υψηλότερους, καλύτερους και αποτελεσματικότερους στόχους”.

ΕΥΧΑΡΙΣΤΙΕΣ

Με την ολοκλήρωση της διπλωματικής μου εργασίας, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αγάπη τους, την υποστήριξη τους και την πίστη τους σε εμένα.

Οφείλω να ευχαριστήσω περισσότερο τον επιβλέποντα Καθηγητή μου κ. Δαλάκα Βασίλειο για την υποστήριξη και την αμέριστη προσοχή που μου παρείχε καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Αισθάνομαι πολύ τυχερή που στο διάστημα αυτό είχα πάντα την ηθική και υλική υποστήριξη που χρειάζομαι για να προχωρήσω.

Πίνακας περιεχομένων

Περίληψη	9
Abstract	11
Κατάλογος Εικόνων	12
Κατάλογος Πινάκων	14
Εισαγωγή	15
Κεφάλαιο 1 ^ο : Αυτοματοποίηση Διαδικασιών Τεχνολογίας Πληροφοριών	17
1.1 Έννοια Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	17
1.2 Τρόπος λειτουργίας Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	19
1.3 Πλεονεκτήματα Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	20
1.4 Μειονεκτήματα Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	23
1.5 Πεδία εφαρμογής Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	26
1.6 Προκλήσεις εφαρμογής Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	30
Κεφάλαιο 2 ^ο : Αυτοματοποιημένη Παραμετροποίηση & Διαχείριση Ηλεκτρονικών Υπολογιστών, Υποδομών Τεχνολογίας Πληροφοριών & Εφαρμογών	32
2.1 Συστήματα Διαχείρισης – Διαμόρφωσης Λογισμικού (Software Configuration Management – SCM)	32
2.2 Τάσεις και Μερίδια Αγοράς	33
2.3 Εργαλεία Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών	38
2.3.1 Εργαλείο «CFEngine»	38
2.3.2 Εργαλείο «Rudder»	42
2.3.3 Εργαλείο «Puppet»	44
2.3.4 Εργαλείο «Chef»	48
2.3.5 Εργαλείο «Salt Stack»	51
2.3.6 Εργαλείο «Ansible»	56
2.4 Σύγκριση Δυνατοτήτων - Επιλογή	60
Κεφάλαιο 3 ^ο : Ορισμός του Προβλήματος και Τεχνική Ανάπτυξη	68
3.1 Ορισμός του Προβλήματος	68
3.2 Αρχιτεκτονική Περιβάλλοντος Δοκιμών - Παραδοχές	70
3.2.1 Περιβάλλον Εργασίας του λογισμικού «VMware»	73
3.2.2 Εικονική μηχανή «Linux – Debian 10 Buster»	80
3.2.3 Εικονική μηχανή "Windows 10"	92

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

3.3	Επικοινωνία εργαλείου «Ansible» και εικονικών μηχανών «Windows»	106
Κεφάλαιο 4 ^ο : Μελέτες Περίπτωσης (Case Studies)		108
4.1	Εγκατάσταση λογισμικού «Android Studio» και εργαλείων «SDK» (SDK Tools).....	108
4.2	Εισαγωγή Ηλεκτρονικού Υπολογιστή με λειτουργικό σύστημα «Windows» σε δίκτυο τομέα «Windows Domain Network»	128
4.3	Εξαγωγή Ηλεκτρονικού Υπολογιστή με λειτουργικό σύστημα «Windows» από δίκτυο τομέα «Windows Domain Network»	134
4.4	Εγκατάσταση λογισμικού προσομοίωσης «Simul8»	139
Κεφάλαιο 5 ^ο : Συμπεράσματα και Μελλοντικές Επεκτάσεις.....		143
5.1	Συμπεράσματα	143
5.2	Μελλοντικές Επεκτάσεις	147
Βιβλιογραφία		149
Παραρτήματα.....		161
Παράρτημα Α΄		161
Παράρτημα Β΄		165
Παράρτημα Γ΄		166
Παράρτημα Δ΄		167

Περίληψη

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η αυτοματοποιημένη παραμετροποίηση συστημάτων Ηλεκτρονικών Υπολογιστών, οι οποίοι λειτουργούν με λειτουργικά συστήματα «Windows», χρησιμοποιώντας το εργαλείο «Ansible» και τις δυνατότητές του.

Στο πρώτο κεφάλαιο γίνεται μνεία στην έννοια και τον τρόπο λειτουργίας της αυτοματοποίησης τεχνολογίας πληροφοριών σε ένα γενικότερο πλαίσιο. Αναλύονται πλεονεκτήματα, μειονεκτήματα και περιγράφεται ένα ευρύ φάσμα πεδίων εφαρμογής της αυτοματοποίησης διαδικασιών τεχνολογίας πληροφοριών (IT Process Automation).

Στο δεύτερο κεφάλαιο, περιγράφεται ειδικότερα η αυτοματοποιημένη παραμετροποίηση και διαχείριση Ηλεκτρονικών Υπολογιστών και υποδομών πληροφορικής. Αναλύονται και συγκρίνονται τα δημοφιλέστερα εργαλεία διαχείρισης διαμόρφωσης λογισμικού (Configuration Management Tools).

Στο τρίτο κεφάλαιο παρουσιάζονται όλα τα στάδια της τεχνικής ανάπτυξης του περιβάλλοντος που δημιουργήθηκε για τις ανάγκες της παρούσας διπλωματικής εργασίας. Αναλύεται η παραμετροποίηση του λογισμικού «VMware Workstation», καθώς και των εικονικών μηχανών που δημιουργήθηκαν. Περιγράφεται η δημιουργία και η παραμετροποίηση του μηχανήματος ελέγχου του «Ansible» (Ansible Control machine), σε λειτουργικό σύστημα «Linux» διανομής «Debian», καθώς και των εικονικών μηχανών με λειτουργικό σύστημα «Windows», ούτως ώστε να μπορούν να διαμορφωθούν από το «Ansible».

Στο τέταρτο κεφάλαιο αναλύονται εκτενώς οι μελέτες περίπτωσης εφαρμογής του εργαλείου «Ansible». Αναλύεται η αξιοποίηση των ενοτήτων λογισμικού (modules) του «Ansible», προκειμένου να επιτευχθεί αυτοματοποιημένα και απομακρυσμένα η εγκατάσταση κάθε εξεταζόμενου λογισμικού με τη χρήση ενός «Playbook», από το αρχικό στάδιο της λήψης (download) του έως και τη χρήση του από τον τελικό χρήστη (end user). Επιπλέον, εξετάζεται η αυτοματοποίηση των διαδικασιών εισαγωγής και εξαγωγής ενός υπολογιστή – στόχου σε ένα δίκτυο τομέα (Windows domain network).

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Στο πέμπτο και τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπόνηση της εν λόγω διπλωματικής εργασίας και γίνεται αναφορά σε μελλοντικές επεκτάσεις της χρήσης του «Ansible» στα λειτουργικά συστήματα «Windows».

Λέξεις κλειδιά: Ansible, Windows, Linux, Αυτοματοποιημένη Παραμετροποίηση, Διαχείριση, Διαμόρφωση.

Abstract

The subject of this dissertation is the automated configuration of electronic computers, which operate with "Windows" operating systems, utilizing the "Ansible" tool and its capabilities.

The first chapter refers to the concept and way of operation of information technology automation in a more general context. Benefits, disadvantages are analyzed and a wide range of fields of application for IT Process Automation (IT Process Automation) are described.

In the second chapter, the automated configuration and management of Computers and IT infrastructures are described in particular. The most popular software configuration management tools (Configuration Management Tools) are analyzed and compared.

The third chapter presents all the stages of the technical development of the environment created for the needs of this dissertation. The configuration of the "VMware Workstation" software and the virtual machines that were created, is analyzed. The creation and configuration of the "Ansible control machine", with "Linux Debian distribution" operating system, as well as of the virtual machines with "Windows" operating system, so that they can be configured by "Ansible", are described.

In the fourth chapter, the case studies of the "Ansible" tool are extensively analyzed. The utilization of the software sections (modules) of "Ansible", in order to automatically and remotely achieve the installation of each examined software using a "Playbook", from the initial stage of downloading it to its use by end user, is analyzed. In addition, the automation of the processes of importing and exporting a target computer to a Windows domain network is considered.

The fifth and final chapter presents the conclusions drawn from the elaboration of this dissertation and refers to future extensions of the use of "Ansible" in "Windows" operating systems.

Keywords: Ansible, Windows, Linux, Automated Configuration, Management, Configuration.

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Κατάλογος Εικόνων

Εικόνα 1: Σειρά κατάταξης σε παγκόσμιο επίπεδο των λογισμικών διαχείρισης διαμόρφωσης.	36
Εικόνα 2: Σειρά κατάταξης των λογισμικών διαχείρισης διαμόρφωσης στην Ελλάδα.	38
Εικόνα 3: Απεικόνιση διεπαφής ιστού (web interface) διαχειριστή CFEngine.	41
Εικόνα 4: Απεικόνιση διεπαφής ιστού (web interface) διαχειριστή CFEngine ως προς ένα συγκεκριμένο κόμβο.	42
Εικόνα 5: Αρχιτεκτονική του εργαλείου «Rudder».	44
Εικόνα 6: Αρχιτεκτονική του εργαλείου «Puppet».	47
Εικόνα 7: Αρχιτεκτονική του εργαλείου «Chef»	49
Εικόνα 8: Αρχιτεκτονική του εργαλείου «Salt».....	53
Εικόνα 9: Χρήση των δημοφιλέστερων εργαλείων αυτοματοποιημένης διαχείρισης διαμόρφωσης ανά έτος.	63
Εικόνα 10: Τάση χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης τα τελευταία πέντε (5) έτη.	63
Εικόνα 11: Τάση χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης τα τελευταία πέντε (5) έτη.	64
Εικόνα 12: Σύνολα «pull requests» των δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης στον ιστότοπο «GitHub».	64
Εικόνα 13: Σύγκριση και τάσεις χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης στον ιστότοπο «GitHub».	65
Εικόνα 14: Τάσεις χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης ανά έτος. ...	65
Εικόνα 15: Παγκόσμιο μερίδιο της αγοράς λειτουργικών συστημάτων επιτραπέζιων υπολογιστών.....	68
Εικόνα 16: Περιβάλλον του «VMware Workstation».	74
Εικόνα 17: Δημιουργία και παραμετροποίηση εικονικής μηχανής (VM) στο «VMWare Workstation».	75
Εικόνα 18: Παραμετροποίηση προσαρμογέων δικτύου (Network Adapters).....	76
Εικόνα 19: Χαρακτηριστικά «Windows Server 2016 VM».	78
Εικόνα 20: Χαρακτηριστικά «Windows 10 Professional VM».	79
Εικόνα 21: Χαρακτηριστικά «Linux Debian 10 Buster VM».....	80
Εικόνα 22: Επιτυχής εγκατάσταση του «Ansible».	82

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Εικόνα 23: Δομή «Ansible Vault».	88
Εικόνα 24: Παραγόμενο κρυπτοκείμενο (ciphertext).	90
Εικόνα 25: Έκδοση Λειτουργικού Συστήματος.	94
Εικόνα 26: Έκδοση «PowerShell».	95
Εικόνα 27: Αποτέλεσμα εκτέλεσης του «cmdlet Enable-PSRemoting».	96
Εικόνα 28: Ακροατές «WinRM (WinRM Listeners)	97
Εικόνα 29: Ρυθμίσεις υπηρεσίας «WinRM».	101
Εικόνα 30: Επιτυχής παραμετροποίηση και επικοινωνία μεταξύ του εργαλείου «Ansible» και των λειτουργικών συστημάτων «Windows».	108
Εικόνα 31: Δημιουργία αρχείου «Playbook» για την εγκατάσταση του «Android Studio».	110
Εικόνα 32: Συνολικά αποτελέσματα από την ολοκλήρωση της εκτέλεσης «Playbook».	125
Εικόνα 33: Αναζήτηση των εργαλείων «SDK» (SDK Tools).	125
Εικόνα 34: Εντοπισμός «SDK Tools».	126
Εικόνα 35: Σύνοψη Ρυθμίσεων Λογισμικού «Android Studio» στον τελικό χρήστη (end user).	127
Εικόνα 36: Επιτυχής Αυτοματοποιημένη Εγκατάσταση του «Android Studio» και των «SDK Tools».	127
Εικόνα 37: Επιτυχής εκτέλεση «Playbook» εισαγωγής υπολογιστή στο δίκτυο τομέα (domain network).	132
Εικόνα 38: Active Directory Windows Domain Controller.	133
Εικόνα 39: Επιτυχής εκτέλεση «Playbook» εξαγωγής υπολογιστή από δίκτυο τομέα (domain network).	137
Εικόνα 40: Απενεργοποίηση υπολογιστή που εξήχθη από το δίκτυο τομέα στο Active Directory στον Domain Controller Server.	138
Εικόνα 41: Υπολογιστής σε ομάδα εργασίας των λειτουργικών συστημάτων «Windows».	139

Κατάλογος Πινάκων

Πίνακας 1: Μερίδια αγοράς λογισμικών διαχείρισης διαμόρφωσης σε παγκόσμιο επίπεδο.....	36
Πίνακας 2: Μερίδια αγοράς λογισμικών διαχείρισης διαμόρφωσης στην Ελλάδα.	38
Πίνακας 3: Συγκριτικός πίνακας εργαλείων αυτοματοποιημένης διαχείρισης διαμόρφωσης.....	62

Εισαγωγή

Στις μέρες μας, καθίσταται πλέον όλο και εντονότερα αισθητή η ανάγκη για την αποδοτικότερη λειτουργία των Ηλεκτρονικών Υπολογιστών σε δίκτυα. Είτε μιλάμε για εργασιακά και παραγωγικά περιβάλλοντα, είτε για περιβάλλοντα επιμόρφωσης ή εκπαίδευσης, ο αριθμός των εμπλεκόμενων φυσικών χρηστών είναι ιδιαίτερα μεγάλος. Η ικανοποίηση των διαφορετικών απαιτήσεων που διαμορφώνονται, απαιτεί τη χρήση πολλαπλών εργαλείων και την εφαρμογή ποικίλων πολιτικών από τους διαχειριστές τέτοιων δικτύων.

Η διαχείριση ενός δικτύου, ωστόσο, ίσως να απαιτεί την φυσική παρουσία του διαχειριστή, για την εφαρμογή ορισμένων διαδικασιών με μη αυτοματοποιημένο τρόπο, καθώς αν και κάποιες από αυτές τις εργασίες εκτελούνται απομακρυσμένα, τόσο η υλοποίηση, όσο και η ολοκλήρωση τους, χρήζει περαιτέρω ενεργειών στο επίπεδο του χρήστη και του Ηλεκτρονικού Υπολογιστή - Πελάτη (host). Ένα τέτοιο περιβάλλον συνιστά και ο χώρος εργαστηρίων ηλεκτρονικών υπολογιστών που φιλοξενούν τα Πανεπιστήμια, τα γραφεία και τα τμήματα μιας επιχείρησης ή ενός οργανισμού, τα γραφεία ενός δημόσιου φορέα, ενός υπουργείου, ενός επιτελείου των Ενόπλων Δυνάμεων κλπ.

Σημαντικό μέρος των δραστηριοτήτων που εκτελεί ένας διαχειριστής πληροφοριακών συστημάτων στα περιβάλλοντα που προαναφέρθηκαν, περιλαμβάνουν τη καθημερινή διεξαγωγή πανομοιότυπων διαδικασιών, όπως η ανάπτυξη ή τροποποίηση εικονικών μηχανών (Virtual Machines – VM), η προσαρμογή του λειτουργικού συστήματος ενός υπολογιστή – πελάτη (host), η ενημέρωση πακέτων λογισμικού ή/και λειτουργικών συστημάτων, η εφαρμογή επιδιορθώσεων ασφαλείας (security patches), η αντιγραφή αρχείων, οι επανεκκινήσεις, ο έλεγχος της κατάστασης μιας υπηρεσίας συστήματος κλπ. Όμως, όλα αυτά υπόκεινται σε ανθρώπινη παρέμβαση, με αποτέλεσμα να αυξάνονται οι πιθανότητες τυχόν αστοχιών, που μπορούν να θέσουν σε κίνδυνο την τυποποίηση μιας υποδομής πληροφορικής, καθώς και να προκληθούν σημαντικές και ζημιογόνες καθυστερήσεις. Πέραν των ανωτέρω, οι μονότονες επαναλαμβανόμενες εργασίες που αναγκάζονται να εκτελούν οι διαχειριστές πληροφοριακών συστημάτων, οδηγούν στην έλλειψη κινήτρου και εν τέλει στη χαμηλή παραγωγικότητα τους,

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

εμποδίζοντας παράλληλα την επεκτασιμότητα των υπηρεσιών και της υποδομής πληροφορικής στο σύνολό της.

Για την αντιμετώπιση των παραπάνω, μπορεί να εφαρμοσθούν σύγχρονες μέθοδοι και εργαλεία, τα οποία προάγουν την αυτοματοποίηση διαδικασιών, οι οποίες μπορεί εκ πρώτης όψης να φαντάζουν απλές, πλην όμως καταναλώνουν μεγάλο μέρος του διαθέσιμου παραγωγικού χρόνου του διαχειριστή μιας υποδομής πληροφορικής. Με τη χρήση τέτοιων εργαλείων, εκτός από την αύξηση της παραγωγικότητας του προσωπικού, μειώνονται αισθητά και οι πιθανότητες αστοχιών και σφαλμάτων που οφείλονται στην ανθρώπινη παρέμβαση. Επιπλέον, βελτιώνεται η ταχύτητα υλοποίησης των εκάστοτε απαιτήσεων, ενισχύεται η ασφάλεια ενός οργανισμού ή μιας επιχείρησης και εξοικονομούνται τόσο ανθρώπινοι όσο και οικονομικοί πόροι.

Κεφάλαιο 1^ο: Αυτοματοποίηση Διαδικασιών Τεχνολογίας Πληροφοριών

1.1 Έννοια Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

Ο «αυτοματισμός» είναι ένας ευρύτερος όρος, συχνά συνδυασμένος ή συνδεδεμένος με τον αυτοματισμό των επιχειρηματικών εργασιών. Μια αυτοματοποιημένη ροή εργασίας πληροφορικής μπορεί να επιτελέσει μια αμιγώς εργασία ενός τμήματος Πληροφορικής σε παραγωγικό περιβάλλον (π.χ. ενός οργανισμού, μιας επιχείρησης κλπ), όπως η παροχή πρόσθετου χώρου αποθήκευσης σε μία εικονική μηχανή (Virtual Machine - VM) ή μια οποιαδήποτε «επιχειρηματική» εργασία, όπως η δημιουργία ενός νέου λογαριασμού χρήστη στο εταιρικό σύστημα αλληλογραφίας.

Ο «αυτοματισμός διαδικασιών πληροφορικής» (IT Process Automation - ITPA) είναι η διαδικασία δημιουργίας λογισμικού και συστημάτων με σκοπό την αντικατάσταση επαναλαμβανόμενων διαδικασιών και τη μείωση της παρέμβασης με μη αυτοματοποιημένο τρόπο (manual). Μέσω της αυτοματοποίησης διαδικασιών που προηγουμένως απαιτούσαν ανθρώπινη παρέμβαση, επιτυγχάνεται η επιτάχυνση και η βελτιστοποίηση της υποδομής, καθώς και των εφαρμογών πληροφορικής. Για την επίτευξη αυτοματισμών πληροφορικής, χρησιμοποιείται λογισμικό που συμβάλλει στη ρύθμιση και την επανάληψη οδηγιών, διαδικασιών ή πολιτικών που εξοικονομούν χρόνο και εργατώρες, με αποτέλεσμα το ανθρώπινο δυναμικό του τμήματος Πληροφορικής μιας επιχείρησης ή ενός οργανισμού να είναι διαθέσιμο και να εργάζεται με πιο παραγωγικό και στρατηγικό τρόπο.

Ο αυτοματισμός μιας διεργασίας βελτιώνει τις ροές εργασίας, σε οργανισμού ή επιχειρήσεις, όπου συνήθως η ίδια εργασία ή σειρά εργασιών διεξάγεται επανειλημμένα. Η αυτοματοποίηση μιας επιχειρησιακής διαδικασίας (Business Process Automation) δεν είναι παρά η εφαρμογή του αυτοματισμού πληροφορικής για την επίτευξη συγκεκριμένων στόχων, όπως η αύξηση της παραγωγικότητας των εργαζομένων ή μείωση του κόστους εργασίας. Μερικοί επαγγελματίες αναφέρονται στην έννοια του «αυτοματισμού πληροφορικής», χρησιμοποιώντας τον όρο «αυτοματισμός υπηρεσιών», ταυτίζοντας επί της ουσίας λειτουργικά τους δύο όρους.

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Επιπλέον, αξίζει να αναφερθεί ότι ο «αυτοματισμός των διαδικασιών πληροφορικής» (IT Process Automation - ITPA), διαφέρει από την «ενορχήστρωση» (Orchestration), αλλά συνήθως οι όροι χρησιμοποιούνται από κοινού. Ο «αυτοματισμός» εκτελεί μια εργασία επανειλημμένα χωρίς ανθρώπινη παρέμβαση. Η «ενορχήστρωση» αποτελεί μια ευρύτερη έννοια, όπου ο χρήστης συντονίζει τις αυτοματοποιημένες εργασίες σε μια συνεκτική διαδικασία ή ροή εργασίας για το τμήμα Πληροφορικής και τα υπόλοιπα τμήματα της επιχείρησης. Η αυτοματοποίηση διαδικασιών πληροφορικής, δηλαδή, διευκολύνει την ενορχήστρωση και ενσωμάτωση εργαλείων, ατόμων και διαδικασιών μέσω αυτοματοποιημένων ροών εργασίας. Οι εφαρμογές λογισμικού «ITPA» μπορούν να προγραμματισθούν για την εκτέλεση οποιουδήποτε επαναλαμβανόμενου μοτίβου, εργασίας ή επιχειρηματικής ροής εργασίας που κάποτε απαιτούσε τον χειροκίνητο χειρισμό από το ανθρώπινο δυναμικό μιας επιχείρησης.

Για παράδειγμα, ένας διαχειριστής συστημάτων πληροφορικής επιτρέπει την κατανομή του φόρτου εργασίας (load balancing), παραδείγματος χάρη, με την αυτοματοποιημένη δημιουργία στιγμιότυπων εικονικών μηχανών (Virtual Machine Instances), την εγκατάσταση λειτουργικών συστημάτων και την παροχή αποθηκευτικού χώρου και πρόβλεψης χρήσης αυτού (provisioning). Για να επιτευχθεί η αυτοματοποίηση των προαναφερόμενων διαδικασιών, ο διαχειριστής συγκεντρώνει και ενορχηστρώνει τις εργασίες αυτοματοποίησης σε μια ροή εργασίας με μια συγκεκριμένη σειρά λειτουργιών για κάθε εργασία. Η ενορχήστρωση μπορεί, επίσης, να περιλαμβάνει χορήγηση δικαιωμάτων, επιβολή ρόλων, και άλλα.

Τα τμήματα Πληροφορικής σε οργανισμούς κάθε μεγέθους και κλάδου μπορούν να αξιοποιήσουν τις τεχνολογίες αυτοματοποίησης για τον καθορισμό κριτηρίων και αποτελεσμάτων σχεδόν για οποιαδήποτε εργασία ή διαδικασία που εκτελείται. Όταν λαμβάνει χώρα ένα συμβάν συστήματος (System Event), ενεργοποιείται και εκτελείται η κατάλληλη ροή εργασίας για την αυτόματη αντιμετώπιση και αποκατάσταση του προβλήματος¹.

¹ <https://searchitoperations.techtarget.com/definition/IT-automation>·
https://en.wikipedia.org/wiki/IT_process_automation·
<https://www.redhat.com/en/topics/automation/whats-it-automation>

1.2 Τρόπος λειτουργίας Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

Η αυτοματοποίηση διαδικασιών πληροφορικής βασίζεται σε εργαλεία λογισμικού για τον ορισμό και τη διεξαγωγή μιας καθορισμένης σειράς λεπτομερών ενεργειών που υλοποιούνται με μη αυτοματοποιημένο τρόπο (χειροκίνητα) ή από κάποιο εξωγενή παράγοντα.

Ο αυτοματισμός πληροφορικής αντικαθιστά μια σειρά ενεργειών και αποκρίσεων μεταξύ ενός διαχειριστή και ενός περιβάλλοντος πληροφορικής. Για παράδειγμα, η διεπαφή γραμμής εντολών (Command Line Interface - CLI) «Microsoft Windows PowerShell», που εμπεριέχεται στα λειτουργικά συστήματα «Microsoft Windows», συνδυάζει εντολές που εκτελούνται σε αυτό (cmdlets), μεταβλητές και άλλα στοιχεία σε μία δέσμη – σενάριο εντολών (script) μέσω της διεπαφής γραμμής εντολών (Command Line Interface - CLI) με σκοπό την παρακολούθηση και παροχή μιας «εικονικής μηχανής» (VM) ή την εφαρμογή μιας διαδικασίας δημιουργίας αντιγράφων ασφαλείας (Backup Operations). Μια πιο σύνθετη υλοποίηση αυτοματοποίησης πληροφορικής, μπορεί να επιτευχθεί συνδυάζοντας δέσμες- σενάρια εντολών (scripts) σε μια σειρά.

Οι διαδικασίες αυτοματοποίησης που εστιάζουν σε ένα συγκεκριμένο εύρος και οι οποίες αντικαθιστούν μια εργασία που πρέπει να εκτελεί συχνά ένας διαχειριστής, κρίνονται ιδιαίτερα αποτελεσματικές. Αυτοματοποιώντας μια ενέργεια που λαμβάνει χώρα πολλές φορές κατά τη διάρκεια της ημέρας, αυξάνεται σημαντικά ο χρόνος που μπορεί ένας διαχειριστής να διαθέσει για άλλες εργασίες που απαιτούν δεξιότητες λήψης αποφάσεων και αξιολόγησης.

Τα εργαλεία αυτοματοποίησης μιας υποδομής πληροφορικής μιας επιχείρησης ή ενός οργανισμού, ενεργοποιούν διαδικασίες οι οποίες ανταποκρίνονται στις ανάγκες και στις υπάρχουσες λειτουργικές συνθήκες του εν λόγω περιβάλλοντος πληροφορικής. Επιπλέον, επικουρούν την επίβλεψη της διαμόρφωσης συστημάτων, λογισμικού και άλλων στοιχείων της υποδομής, συμβάλλουν στην αναγνώριση μη εξουσιοδοτημένων ή μη αναμενόμενων αλλαγών και εκδηλώνουν αυτόματα διορθωτικές ενέργειες. Για παράδειγμα, εάν μια εργασία σταματήσει

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

να ανταποκρίνεται, ενεργοποιούνται τα αυτοματοποιημένα βήματα για την επανεκκίνηση σε διαφορετικό εξυπηρετητή, ο οποίος είναι διαθέσιμος για να την εκτελέσει².

1.3 Πλεονεκτήματα Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

Χάρη στον αυτοματισμό, τα περιβάλλοντα πληροφορικής μπορούν να επεκταθούν πιο γρήγορα, πιο αποτελεσματικά και με λιγότερα σφάλματα, καθώς και να ανταποκρίνονται καλύτερα στις επιχειρηματικές ανάγκες. Ένα αυτοματοποιημένο περιβάλλον μπορεί να μειώσει τους χρόνους παροχής μιας υπηρεσίας πληροφορικής, από εβδομάδες σε λιγότερο από μια ημέρα. Συνεπώς, όλο και περισσότερο, η αυτοματοποίηση πληροφορικής καθίσταται αναγκαία για τις επιχειρήσεις και τους οργανισμούς, προκειμένου να διαχειρίζονται το περίπλοκο σύγχρονο τεχνολογικό τοπίο. Πέραν αυτού, οι διαχειριστές λειτουργιών και συστημάτων πληροφορικής, καλούνται να αυξήσουν την απόδοσή τους έχοντας στη διάθεσή τους λιγότερο ανθρώπινο δυναμικό. Ενδεικτικά, οι αυτοματισμοί πληροφορικής προσφέρουν τα παρακάτω οφέλη³:

➤ **Μείωση κόστους**

Με τη χρήση αυτοματισμών πληροφορικής, απαιτούνται λιγότερες ώρες εργασίας για την ολοκλήρωση εργασιών ρουτίνας. Ο αυτοματισμός μπορεί, επίσης, να αυξήσει την παραγωγικότητα, όπως θα αναλυθεί και περαιτέρω στη συνέχεια, να μειώσει το κόστος της υποδομής βελτιστοποιώντας τη χρήση πόρων και να μειώσει τη ζημία που οφείλεται σε ανθρώπινα λάθη. Η αυτοματοποίηση επαναλαμβανόμενων λειτουργικών εργασιών και διαδικασιών, όπως η εγκατάσταση και χρήση εφαρμογών και η παροχή διαφόρων υπηρεσιών, η διαχείριση αλλαγών και η διαχείριση ενημερώσεων κώδικα, μπορούν να καταστήσουν τις λειτουργίες πληροφορικής περισσότερο αποτελεσματικές, με λιγότερα σφάλματα και παράλληλα να συνεισφέρουν στην εξοικονόμηση οικονομικών πόρων.

²<https://www.vmware.com/topics/glossary/content/it-automation>
<https://searchitoperations.techtarget.com/definition/IT-automation>

³<https://www.vmware.com/topics/glossary/content/it-automation>
<https://searchitoperations.techtarget.com/definition/IT-automation>
<https://www.redhat.com/en/topics/automation/whats-it-automation>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Επιπρόσθετα, αξίζει να σημειωθεί ότι όχι μόνο ζητείται από τους διαχειριστές λειτουργιών και συστημάτων πληροφορικής να εργασθούν περισσότερο, αλλά τους ζητείται να εκτελούν αυτές τις εργασίες ακόμα πιο γρήγορα και πιο αποτελεσματικά. Κατά συνέπεια, λοιπόν, τα διάφορα εργαλεία και οι τεχνικές αυτοματοποίησης πληροφορικής μπορούν να σταθούν αρωγοί στη βελτίωση της απόδοσης τους, χωρίς να απαιτείται η ενίσχυση με επιπλέον έμπυχο εργατικό δυναμικό.

➤ **Ακρίβεια – Ελαχιστοποίηση Σφαλμάτων**

Ένας διαχειριστής πληροφοριακών συστημάτων, κατά την άσκηση των καθηκόντων του, ενδέχεται να κάνει κάποιο σφάλμα κατά την πληκτρολόγηση σε ένα περιβάλλον γραμμής εντολών (Command Line Interface – CLI), να επιλέξει τυχόν λανθασμένη ρύθμιση διαμόρφωσης σε έναν εξυπηρετητή (server), να παραβλέψει ένα βασικό βήμα σε μια σύνθετη εργασία, είτε γενικότερα να προβεί σε άλλα σφάλματα ή παραλείψεις. Τα σφάλματα αναγκάζουν τους διαχειριστές να αφιερώσουν επιπλέον χρόνο για τη διόρθωσή τους, επαναλαμβάνοντας τις όποιες διαδικασίες προκειμένου να επιτευχθεί το σωστό αποτέλεσμα. Ο αυτοματισμός πληροφορικής επιτρέπει σε έναν επαγγελματία πληροφορικής να κατασκευάσει μια αποδεδειγμένη, στοχευμένη ακολουθία λειτουργιών που μπορεί να εκτελεστεί αμέτρητες φορές με τον ίδιο τρόπο. Συνεπώς, ελαχιστοποιείται η πιθανότητα σφαλμάτων που οφείλονται στον ανθρώπινο παράγοντα, αυξάνεται η αξιοπιστία της υποδομής και επιτυγχάνεται η απρόσκοπτη διαθεσιμότητα αυτής.

➤ **Αυξημένη παραγωγικότητα – Ταχύτερες λειτουργίες**

Αυτοματοποιώντας επαναλαμβανόμενες και χρονοβόρες εργασίες, το προσωπικό πληροφορικής αξιοποιεί βέλτιστα το χρόνο του, έχοντας τη δυνατότητα να καταναλώνει το μεγαλύτερο μέρος αυτού σε εργασίες υψηλότερου επιπέδου. Η αυτοματοποίηση των ροών εργασίας εξαλείφει τη μη αυτόματη εργασία, συμπεριλαμβανομένων των χειροκίνητων δοκιμών. Συνεισφέρει στην αύξηση της παραγωγής και στην αποτελεσματικότερη απασχόληση των εργαζομένων, παρέχοντας τους τη δυνατότητα να

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

εστιάζουν στην εκπόνηση σημαντικότερου έργου. Ταυτόχρονα δε, επιτυγχάνεται η αύξηση της ημερήσιας απόδοσης τους.

Ο αυτοματισμός πληροφορικής μπορεί να επιταχύνει σημαντικά τις λειτουργίες του κέντρου δεδομένων (data center) και του υπολογιστικού νέφους (cloud), μειώνοντας τους χρόνους παροχής υπηρεσιών και πόρων από εβδομάδες σε ώρες. Οι λειτουργίες πληροφορικής απαιτούν σημαντικό αριθμό διαφορετικών εργασιών. Ένας διαχειριστής πληροφοριακών συστημάτων μπορεί να εκτελέσει κάθε εργασία με μη αυτοματοποιημένο τρόπο, αλλά οι σύγχρονες επιχειρηματικές απαιτήσεις ασκούν εξαιρετική πίεση στο προσωπικό πληροφορικής προκειμένου αυτό να ανταποκριθεί ταχύτατα στις ανάγκες που δημιουργούν οι αυξημένου μεγέθους και πολυπλοκότητας υποδομές. Το ανθρώπινο δυναμικό δεν μπορεί εύκολα να ανταπεξέλθει σε ένα περιβάλλον έντονης πίεσης, ούτε να ανταποκριθεί σε ένα ιδιαίτερα αυξημένο φόρτο εργασίας μέσα σε ελάχιστο χρόνο, ολοκληρώνοντας το σύνολο των μεμονωμένων εργασιών ρουτίνας που απαιτούνται, οποιαδήποτε στιγμή εντός της ημέρας. Ο αυτοματισμός εξοικονομεί στους διαχειριστές πληροφοριακών συστημάτων πολύτιμο χρόνο που σπαταλάται σε εργασίες ρουτίνας, ώστε να μπορούν να εστιάζουν σε έργα προστιθέμενης αξίας για την επιχείρηση, όπως η βελτιστοποίηση υποδομής πληροφορικής και ο πειραματισμός με πολλά υποσχόμενες νέες τεχνολογίες και προϊόντα.

➤ **Ασφάλεια**

Οι αυτοματοποιημένες διαδικασίες απαιτούν λιγότερους φυσικούς χρήστες να παρακολουθούν και να διαχειρίζονται ευαίσθητες πληροφορίες, γεγονός που μειώνει την πιθανότητα τυχόν παραβίασης. Επιπλέον, ο αυτοματισμός πληροφορικής μπορεί να χρησιμοποιηθεί για να υποβοηθήσει τις ομάδες πληροφορικής να παρακολουθούν τον τρόπο με τον οποίο το σύστημα ανταποκρίνεται σε διάφορα περιστατικά ασφάλειας που μπορεί να προκύψουν, έχοντας παράλληλα τη δυνατότητα να επέμβουν μόνο όταν και εφόσον απαιτηθεί.

➤ **Τυποποίηση**

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Παρατηρείται πολύ συχνά το φαινόμενο διαφορετικοί διαχειριστές πληροφοριακών συστημάτων να εκτελούν την ίδια εργασία με διαφορετικούς τρόπους, και αρκετές φορές ακόμη και ο ίδιος διαχειριστής να χειρίζεται μια εργασία διαφορετικά σε μια δεδομένη χρονική στιγμή σε σύγκριση με κάποια άλλη. Για την απρόσκοπτη εταιρική διακυβέρνηση ή τη διακυβέρνηση ενός οργανισμού και προκειμένου αυτή να είναι αποτελεσματική θα πρέπει να επιτυγχάνεται συμμόρφωση με τους επιβαλλόμενους κανόνες και διαδικασίες λειτουργίας. Η υιοθέτηση μιας στρατηγικής αυτοματισμού πληροφορικής αποδεικνύει την δέουσα συνέπεια στις επιβαλλόμενες και ισχύουσες διαδικασίες λειτουργίας, ανεξάρτητα από τον εκάστοτε διαχειριστή της υποδομής και σε οποιαδήποτε χρονική στιγμή.

1.4 Μειονεκτήματα Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

Αν και ο αυτοματισμός έχει πολλά πλεονεκτήματα, όπως αναφέρθηκε και προηγουμένως, υπάρχουν μερικά σημαντικά στοιχεία που χρήζουν προσοχής. Όπως καθετί, έτσι και οι μέθοδοι, οι τεχνικές και τα εργαλεία αυτοματισμού συστημάτων πληροφορικής, έχουν αδυναμίες οι οποίες κρίνεται χρήσιμο να αναλυθούν. Άλλωστε κάθε διαχειριστής λειτουργιών και συστημάτων πληροφορικής θα πρέπει να έχει μία πλήρη και σαφή εικόνα, ώστε να είναι σε θέση να αξιολογήσει κατάλληλα την επιλογή ή μη υλοποίησης αυτοματισμών στην υποδομή την οποία υποστηρίζει. Ορισμένα μειονεκτήματα παρατίθενται παρακάτω⁴:

➤ Οι συνέπειες των σφαλμάτων

Ο αυτοματισμός μειώνει σημαντικά την πιθανότητα ανθρώπινου λάθους. Ταυτόχρονα όμως, επειδή ο αυτοματισμός προσδίδει έναν ταχύτατο χαρακτήρα στις διαδικασίες που εκτελούνται, τυχόν «αυτοματοποιημένα σφάλματα» είναι πιθανό να προκαλέσουν πολύ μεγαλύτερη ζημία από αυτή που ίσως να προκαλούσαν τα σφάλματα ενός φυσικού διαχειριστή πληροφοριακών συστημάτων. Αυτό καθιστά μείζονος σημασίας τη σωστή

⁴<https://www.vmware.com/topics/glossary/content/it-automation>
<https://searchitoperations.techtarget.com/definition/IT-automation>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

ρύθμιση και παραμετροποίηση των εργαλείων αυτοματισμού που θα επιλεγούν, καθώς επίσης και τη δοκιμή αυτών πριν από την εφαρμογή τους.

Παρόλο που η δυνατότητα αμέτρητων επαναλήψεων μιας διαδικασίας, χωρίς απόκλιση σε κάθε επανάληψη, αποτελεί ένα από τα σημαντικότερα πλεονεκτήματα της αυτοματοποίησης διαδικασιών πληροφοριακών συστημάτων, μπορεί υπό συνθήκες να καταστεί επιζήμια. Τα λάθη και οι παραλείψεις κωδικοποιούνται εύκολα σε μια αυτοματοποιημένη διαδικασία, την οποία το εργαλείο αυτοματισμού θα εκτελέσει τόσο γρήγορα και αποτελεσματικά, όπως ακριβώς θα έπραττε και με τα αντίστοιχα ορθά βήματα. Εάν ο διαχειριστής αυτοματοποιήσει μια πολύπλοκη ακολουθία συμβάντων και παραλείψει ένα βασικό βήμα ή ορίσει λανθασμένα μια μεταβλητή, τότε αυτό το σφάλμα θα επαναλαμβάνεται συνεχώς, έως ότου εντοπιστεί και αποκατασταθεί από το διαχειριστή. Επομένως, οι αυτοματοποιημένες διαδικασίες δοκιμών και ελέγχου πρέπει να αποτελούν μέρος μιας στρατηγικής αυτοματοποίησης πληροφορικής, για την αποφυγή και την πρόληψη τέτοιων συμβάντων.

➤ **Κόστη Αρχικής Επένδυσης**

Αν και ο αυτοματισμός συστημάτων πληροφορικής συνεισφέρει αποτελεσματικά στην εξοικονόμηση οικονομικών πόρων, συνήθως απαιτείται μια σημαντική αρχική επένδυση για την αγορά και την παραμετροποίηση των λογισμικών αυτοματισμού. Μια επιχείρηση πρέπει να εξετάσει προσεκτικά τη στρατηγική αυτοματοποίησης των διαδικασιών πληροφορικής, έτσι ώστε να βεβαιωθεί ότι οι διαδικασίες που επιθυμεί να αυτοματοποιηθούν θα σημειώσουν έναν υψηλό δείκτη απόδοσης επένδυσης (Return of Investment - ROI). Οι εργασίες ρουτίνας που καταναλώνουν σημαντικό μέρος από τον διαθέσιμο χρόνο εργασίας του προσωπικού πληροφορικής χρήζουν αυτοματοποίησης, ενώ αντίθετα διαδικασίες που εκτελούνται σπάνια όχι.

➤ **Δυσκολίες Προσαρμογής**

Αρκετά συχνά διάφορες επιχειρήσεις ή οργανισμοί αναγκάζονται για διάφορες αιτίες να τροποποιήσουν τις διαδικασίες που εφαρμόζονται από το τμήμα Πληροφορικής τους. Στην περίπτωση αυτή, απλά και μόνο η χρήση ενός εργαλείου αυτοματοποίησης των

διαδικασιών δεν σημαίνει ότι θα οδηγήσει στην επίτευξη της επιθυμητής αλλαγής. Αυτό εξηγείται από το γεγονός ότι πολλά εργαλεία αυτοματισμού έχουν κατασκευαστεί για την κάλυψη συγκεκριμένων απαιτήσεων ή η εφαρμογή τους περιορίζεται σε συγκεκριμένες λειτουργίες. Επιπρόσθετα, οι διαδικασίες αλλάζουν και σε συνάρτηση με το χρόνο, καθώς η υποδομή πληροφορικής τους μπορεί να επεκτείνεται, να αλλάζει λόγω και της εξέλιξης των υφιστάμενων τεχνολογιών, με αποτέλεσμα να καθίσταται υποχρεωτική η εφαρμογή των βέλτιστων πρακτικών.

Οι αυτοματοποιημένες διαδικασίες παραμένουν στατικές, έως ότου κάποιος αποφασίσει να τις αλλάξει. Οι οργανισμοί πρέπει να έχουν μια καθορισμένη ροή εργασίας για να ενημερώνουν και να επικυρώνουν εκ νέου τις διαδικασίες αυτοματοποίησης, συμπεριλαμβανομένης της «πειθαρχημένης» (disciplined) αυτοματοποίησης εκδόσεων που παρακολουθεί τον τρόπο με τον οποίο οι εργασίες μεταλλάσσονται με την πάροδο του χρόνου.

➤ **Ολοκλήρωση και διαλειτουργικότητα**

Τα εργαλεία αυτοματοποίησης διαδικασιών πληροφορικής πρέπει να είναι συμβατά με συστήματα, λογισμικό και άλλα στοιχεία σε δυνητικά διαφορετικά περιβάλλοντα πληροφορικής. Στην ιδανική περίπτωση, ένα εργαλείο αυτοματισμού θα πρέπει να ενσωματωθεί με εργαλεία ενορχήστρωσης υψηλότερου επιπέδου για να μπορεί συγκεντρώσει εργασίες υπό ελεγχόμενες ροές εργασίας.

➤ **Λανθασμένη Σχεδίαση της Στρατηγικής**

Ενώ ο αυτοματισμός εξοικονομεί χρόνο, απαιτεί από τους διαχειριστές πληροφοριακών συστημάτων να σχεδιάσουν και να ερευνήσουν προσεκτικά, κάθε εργασία που είναι απαραίτητη για την προβλεπόμενη ροή εργασίας και στη συνέχεια να μεταφράσουν σωστά αυτά τα βήματα σε κάποια μέθοδο αυτοματισμού, για να επιτύχουν την επιθυμητή τελική κατάσταση. Μια εταιρεία μπορεί να διορίσει έναν ή περισσότερους διαχειριστές αυτοματισμού πληροφορικής, αντικαθιστώντας ή συμπληρώνοντας το ρόλο των διαχειριστών πληροφοριακών συστημάτων. Η αυτοματοποίηση των διαδικασιών πληροφορικής μπορεί να έχει τα αντίθετα από τα αναμενόμενα αποτελέσματα, εάν

καταστεί απλά και μόνο ένας στόχος χωρίς να έχουν ληφθεί υπόψη η επιθυμητή απόδοση της επένδυσης, η δυναμική του τμήματος Πληροφορικής της επιχείρησης ή του οργανισμού, καθώς και η ανάλυση σε βάθος των επιχειρησιακών διαδικασιών που θα πρέπει να εξυπηρετηθούν. Σε όλες τις περιπτώσεις, το κλειδί της επιτυχίας για την αποφυγή προβλημάτων είναι η επίδειξη ιδιαίτερης προσοχής και η ενδελεχής μελέτη για την ανάπτυξη της κατάλληλης και αποτελεσματικής στρατηγικής αυτοματοποίησης των διαδικασιών πληροφορικής. Ο αυτοματισμός αποτελεί ένα ισχυρό εργαλείο, αλλά είναι εξίσου καλό με μία ισχυρή ομάδα που το αξιοποιεί.

1.5 Πεδία εφαρμογής Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

Είναι γεγονός ότι οι σημερινές ομάδες λειτουργιών πληροφορικής (IT Operations – IT Ops) διαχειρίζονται συνεχώς μεταβαλλόμενες, περίπλοκες αρχιτεκτονικές υποδομών πληροφορικής, οι οποίες βασίζονται σε ποικίλες πλατφόρμες και ιδιόμορφες τεχνολογίες.

Επιπρόσθετα, καθώς οι οργανισμοί υιοθετούν ψηφιακές επιχειρηματικές πρακτικές, τα τμήματα Πληροφορικής τους καλούνται να προσφέρουν επεκτάσιμη, κατ' απαίτηση υποδομή για την υποστήριξη προγραμματιστών, γραμμών επιχειρηματικών δραστηριοτήτων, συνεχή και απρόσκοπτη υποστήριξη ομάδων χρηστών ηλεκτρονικών υπολογιστών, κλπ., ενώ ταυτόχρονα οφείλουν να επιτυγχάνουν ασφάλεια, συμμόρφωση και αξιοπιστία.

Κατά συνέπεια, οι ομάδες λειτουργιών πληροφορικής (IT Ops), αναγνωρίζουν πλέον την ανάγκη υιοθέτησης μιας νέας προσέγγισης αναφορικά με τον τρόπο που παρέχουν τις υπηρεσίες τους στον οργανισμό ή την επιχείρηση στην οποία απασχολούνται. Για την επίτευξη του στόχου αυτού χρησιμοποιούνται σύγχρονα, ενοποιημένα εργαλεία διαχείρισης και αυτοματισμού, προκειμένου να υποστηρίξουν αποτελεσματικά τις όποιες διαδικασίες και λειτουργίες του εκάστοτε οργανισμού ή επιχείρησης.

Καθίσταται προφανές, λοιπόν, πως η αυτοματοποίηση λειτουργιών πληροφορικής αποτελεί ένα ισχυρό εργαλείο που συμβάλλει στην επέκταση μιας επιχείρησης ή ενός οργανισμού, προσφέρει σημαντική εξοικονόμηση κόστους και επιτρέπει στο προσωπικό πληροφορικής να

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

επικεντρωθεί σε καθήκοντα περισσότερο στρατηγικού (Strategic) παρά διοικητικού (Administrative) χαρακτήρα.

Με την άνοδο και την ανάπτυξη των εικονικοποιημένων δικτύων (virtualized networks) και των υπηρεσιών υπολογιστικού νέφους (cloud) που απαιτούν ταχεία και αυξημένη διαθεσιμότητα (availability), ο αυτοματισμός συνιστά απαραίτητο στοιχείο του στρατηγικού σχεδιασμού μιας παραγωγικής μονάδας, έτσι ώστε να υποστηρίξει τις ομάδες πληροφορικής να παράσχουν βελτιωμένες υπηρεσίες με ταχύτητα, συνέπεια, συνοχή και ασφάλεια.

Ένα ευρύ φάσμα λειτουργιών μιας επιχείρησης ή ενός οργανισμού, όπως κάποιες λειτουργίες ενός κέντρου δεδομένων (data center), μιας υποδομής υπολογιστικού νέφους (cloud infrastructure) ή ενός τμήματος τεχνικής υποστήριξης (IT technical support), μπορεί να αυτοματοποιηθεί, με αποτέλεσμα τη μεγιστοποίηση της ταχύτητας εκτέλεσης τους, καθώς και την αυξημένη παραγωγικότητα των τμημάτων Πληροφορικής της συγκεκριμένης επιχείρησης ή του οργανισμού.

Οι πιθανές εφαρμογές αυτοματισμού αγγίζουν σχεδόν το άπειρο. Οι διαχειριστές πληροφοριακών συστημάτων μπορούν να αξιοποιήσουν, συνεπώς, την αυτοματοποίηση πληροφορικής για διάφορες εργασίες όπως⁵:

➤ **Διαχείριση συμβάντων (Incidents Management)**

Παρόλου που οι επιχειρήσεις και οι οργανισμοί δεν μπορούν να αποφύγουν το σύνολο των σοβαρών περιστατικών, ο αυτοματισμός πληροφορικής μπορεί να βοηθήσει σημαντικά στην αντιμετώπισή τους όταν αυτά εν τέλει συμβούν. Η χρήση αυτοματισμού για την αντιμετώπιση σοβαρών περιστατικών συμβάλλει στην αποκατάσταση των παρεχόμενων υπηρεσιών ταχύτερα και με λιγότερα σφάλματα. Επιπλέον, ο αυτοματισμός πληροφορικής επιτρέπει στις εταιρείες να μειώσουν τη διάρκεια των

⁵ <https://www.redhat.com/en/resources/ansible-continuous-integration-delivery-whitepaper>
<https://www.redhat.com/en/engage/automated-enterprise-ebook-20171107>
<https://www.redhat.com/cms/managed-files/ma-network-automation-for-everyone-e-book-f19707wg-201910-en.pdf>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

συμβάντων αυτών, αλλά και το κόστος που θα προκαλέσουν τόσο στις ίδιες, όσο και στους πελάτες τους.

➤ **Ανάπτυξη & Διανομή Εφαρμογών (Application Deployment)**

Οργανισμοί που χρησιμοποιούν παραδοσιακούς ή συνεχείς τρόπους ενσωμάτωσης (continuous integration) και ανάπτυξης εφαρμογών, αυτοματοποιώντας βασικές εργασίες και δυνατότητες, ιδιαίτερα κατά τη διάρκεια των δοκιμών που εκτελούν, μπορούν να επωφεληθούν ώστε να αναπτύξουν επιτυχώς τις εφαρμογές τους. Ο αυτοματισμός βοηθά τις εταιρείες και τους οργανισμούς να προχωρήσουν από τη δέσμευση και την κατασκευή έως τις δοκιμές, και την ανάπτυξη με πιο συστηματικό τρόπο, βελτιώνοντας την αποδοτικότητα και την απόδοση των εφαρμογών τους και μειώνοντας τις πιθανότητες για ανθρώπινη αστοχία.

Χρησιμοποιώντας διαδικασίες αυτοματοποίησης πληροφορικής, οι οργανισμοί μπορούν να αναπτύξουν εφαρμογές με λιγότερες πιθανότητες σφαλμάτων, να διαμορφώσουν τις απαραίτητες υπηρεσίες από την αρχή, καθώς επίσης και να λαμβάνουν τις εφαρμογές και τα αντικείμενά (artifacts) τους. Για παράδειγμα, εργασίες που έχουν τεκμηριωθεί και αποθηκευτεί σε ένα αποθετήριο (repository) μπορούν να ανακτηθούν κατόπιν κάποιου αιτήματος και να διατίθενται μέσω μιας κοινής, διαφανούς προσέγγισης σύμφωνα με την οποία όλα τα μέλη του προσωπικού πληροφορικής τους θα έχουν πρόσβαση σε αυτές.

➤ **Ασφάλεια & Συμμόρφωση (Security & Compliance)**

Οι διαχειριστές συστημάτων πληροφορικής μπορούν να χρησιμοποιήσουν τον αυτοματισμό πληροφορικής για να καθορίσουν και να επιβάλουν πολιτικές ασφάλειας, συμμόρφωσης και διαχείρισης κινδύνων, καθώς και για την αποκατάσταση τυχόν ζητημάτων δημιουργώντας αυτοματοποιημένα βήματα σε όλες τις υποδομές τους. Ο αυτοματισμός πληροφορικής επιτρέπει στους διαχειριστές να διατηρούν την ασφάλεια στο προσκήνιο των διαδικασιών που τελούνται από τη χρήση της τεχνολογίας πληροφοριών και να κατέχουν έναν πιο ενεργητικό ρόλο σε ό,τι αφορά την ασφάλεια. Η

εφαρμογή τυποποιημένων, αυτοματοποιημένων διαδικασιών ασφάλειας στον κυβερνοχώρο και ροών εργασίας διευκολύνει τη συμμόρφωση και τον έλεγχο.

➤ **Παροχή (Provisioning)**

Η παροχή αφορά στο «χώρο» όπου θα αναπτυχθεί μια υποδομή, είτε δηλαδή σε κάποιο φυσικό μηχάνημα (bare metal) είτε σε ιδιωτικό (private), υβριδικό (hybrid) ή δημόσιο (public) υπολογιστικό νέφος (cloud). Προκειμένου να λειτουργήσουν επιχειρηματικά και επιχειρησιακά πληροφοριακά συστήματα, απαιτείται μια υποδομή, καθώς και τα λοιπά παρελκόμενα για τη δημιουργία της. Αυτό παλαιότερα αφορούσε ικριώματα (racks), φυσικά μηχανήματα και καλώδια σε ένα κέντρο δεδομένων (data center). Πλέον, σε σύγχρονα περιβάλλοντα, όλα τα προαναφερόμενα από κέντρα δεδομένων, δίκτυα και αποθηκευτικούς χώρους έως εικονικές μηχανές κλπ., υλοποιούνται κατά κύριο λόγο, με εικονικοποιημένα στοιχεία. Ουσιαστικά, όλα πλέον σχετίζονται με κάποιο λογισμικό. Οι δυνατότητες του αυτοματισμού παρέχουν επίσης – και κατ’ ανάγκη, - την κωδικοποίηση των διαδικασιών. Αυτό συμβάλλει στην ικανοποίηση των απαιτήσεων της επιχείρησης και στη βελτιωμένη αντίληψη των περιορισμών κόστους και χρόνου που τίθενται.

➤ **Διαχείριση Διαμόρφωσης (Configuration Management)**

Είναι λογικό ότι δεν δημιουργούνται και δεν παρέχονται όλες οι εφαρμογές το ίδιο χρονικό διάστημα, μέσα σε έναν οργανισμό. Απαιτούν διαφορετικές ρυθμίσεις, συστήματα αρχείων, θύρες, χρήστες, κλπ. Αφού αυτοματοποιηθεί η παροχή που αναφέρθηκε παραπάνω, πρέπει το τμήμα Πληροφορικής να είναι σε θέση να καταναείμει ορθά τους πόρους που έχει στη διάθεσή του, ως προς τη διανομή, τη δημιουργία και τη λειτουργία των εκάστοτε εφαρμογών και λογισμικού. Επιπλέον, καθώς αυξάνεται η πολυπλοκότητα, απαιτείται ένας καλύτερος τρόπος καταγραφής (log files, monitoring, documentation) των συστημάτων, των εφαρμογών και γενικότερα της όλης υποδομής, ώστε να επιτευχθεί η αποτελεσματικότερη διαχείριση τους. Για το σκοπό αυτό επιλέγεται ο αυτοματισμός διαδικασιών πληροφορικής. Όσο πιο απλή μορφή έχει η αυτοματοποίηση, ιδίως με τη χρήση απλών δεσμών εντολών (scripts) και πρακτικών «ad hoc» για τη διαχείριση του εκάστοτε συστήματος ή υποδομής, τόσο πιο εύκολο είναι να

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

επιτευχθεί το επιθυμητό αποτέλεσμα, και να καθίστανται κατανοητές όλες διαδικασίες από το σύνολο της ομάδας πληροφορικής ενός οργανισμού.

➤ **Ενορχήστρωση (Orchestration)**

Όσο πιο περίπλοκο είναι ένα πληροφοριακό σύστημα, τόσο πιο περίπλοκη είναι και η διαχείρισή του, κι αυτό διότι εντείνεται η ανάγκη συνδυασμού πολλαπλών αυτοματοποιημένων εργασιών και διαμορφώσεων μεταξύ ομάδων συστημάτων ή μηχανών. Ο συνδυασμός αυτός συνιστά άλλωστε και την έννοια της «ενορχήστρωσης». Ο απαιτούμενος έλεγχος όλων αυτών των ενορχηστρώσεων, πραγματοποιείται με ισχυρές λύσεις αυτοματισμού, οι οποίες επιτρέπουν την απρόσκοπτη και αποτελεσματική παρακολούθηση του συνόλου των λειτουργιών της εκάστοτε υποδομής πληροφορικής. Παράλληλα, εξασφαλίζουν τη δυνατότητα διασύνδεσης του συνόλου των απαιτούμενων στοιχείων με αυτή, αλλά και την αυτόνομη εκτέλεσή τους με ιδιαίτερη ευκολία.

1.6 Προκλήσεις εφαρμογής Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

Η νοοτροπία της χρήσης μη τυποποιημένων μεθόδων, καθώς επίσης και η απροθυμία διάχυσης της γνώσης στο σύνολο του προσωπικού μιας επιχείρησης ή ενός οργανισμού, περιορίζει τα οφέλη της αυτοματοποίησης διαδικασιών τεχνολογίας πληροφοριών. Πολλοί οργανισμοί αυτοματοποιούν έναν αριθμό λειτουργιών πληροφορικής σε περιορισμένους, διαφορετικούς τομείς χρησιμοποιώντας μη ευέλικτες «ad hoc» δέσμες εντολών (scripts) ή ιδιόκτητα εργαλεία παλαιότερης τεχνολογίας για συγκεκριμένες συσκευές. Παρόλο που αυτές οι προσεγγίσεις ενδέχεται να επιταχύνουν συγκεκριμένες λειτουργίες, δεν επεκτείνονται σε διαφορετικούς πόρους και μπορεί να δυσχεράνουν τον διαμοιρασμό εμπειρογνωμοσύνης σε ολόκληρο τον οργανισμό.

Πέραν αυτού, είναι συχνά δύσκολο να ενημερωθούν και να επεκταθούν αυτοί οι τύποι αυτοματισμού, λόγω της συνεχούς εξέλιξης των εφαρμοζόμενων τεχνολογιών και της

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

εμφάνισης νέων απαιτήσεων. Απαιτείται μια συνολική προσέγγιση σε επίπεδο επιχείρησης προκειμένου αυτή να αξιοποιήσει την πλήρη αξία του αυτοματισμού. Μια έξυπνη εφαρμογή του αυτοματισμού, μπορεί για παράδειγμα να συνδέσει τα «απομονωμένα αποθετήρια πληροφοριών» (silos) των διαφόρων τμημάτων μιας επιχείρησης ή ενός οργανισμού. Τα εργαλεία αυτοματισμού διαδικασιών τεχνολογίας πληροφοριών ενισχύουν τη συνεργασία και τη διαφάνεια, συμβάλλουν στη βέλτιστη παροχή εργασίας σε όλες τις επιχειρηματικές μονάδες, καθώς και στην αντιμετώπιση των αυξανόμενων απαιτήσεων και προσδοκιών αυτοεξυπηρέτησης, βελτιώνοντας παράλληλα τον έλεγχο της συμμόρφωσης με τους τιθέμενους κανόνες ασφαλείας.

Η εφαρμογή της αυτοματοποίησης διαδικασιών πληροφορικής δεν εγγυάται τα αναμενόμενα αποτελέσματα. Κανένα εργαλείο, περιβάλλον λογισμικού (framework) ή συσκευή δεν διασφαλίζει ότι ο οργανισμός ή η επιχείρηση θα εξαλείψει σφάλματα, θα βελτιώσει το επίπεδο ασφαλείας ή συμμόρφωσης με τους κανόνες της. Τα μέλη του προσωπικού των τμημάτων Πληροφορικής χρειάζονται σχετικές ικανότητες και δεξιότητες για τη χρήση των εργαλείων αυτοματισμού διαδικασιών πληροφορικής, έτσι ώστε να είναι σε θέση να τυποποιήσουν τις διάφορες διαδικασίες, δημιουργώντας συγκεκριμένα βήματα. Για παράδειγμα, για να χρησιμοποιηθεί η διεπαφή γραμμής εντολών (command line interface) «PowerShell», το προσωπικό πληροφορικής πρέπει να κατανοήσει εκατοντάδες «cmdlet» με σωστή σύνταξη και χρήση της γραμμής εντολών⁶.

⁶ <https://www.redhat.com/en/engage/automated-enterprise-ebook-20171107>
<https://www.redhat.com/cms/managed-files/ma-network-automation-for-everyone-e-book-f19707wg-201910-en.pdf>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Κεφάλαιο 2^ο: Αυτοματοποιημένη Παραμετροποίηση & Διαχείριση Ηλεκτρονικών Υπολογιστών, Υποδομών Τεχνολογίας Πληροφοριών & Εφαρμογών

2.1 Συστήματα Διαχείρισης – Διαμόρφωσης Λογισμικού (Software Configuration Management – SCM)

Η χρήση εικονικών μηχανών (VMs), καθώς και η τακτική διαμόρφωση και χρήση συσκευών δικτύωσης, όπως δρομολογητές (routers), μεταγωγείς (switches) κλπ, έχει επηρεάσει σημαντικά τα πληροφοριακά συστήματα και το ρόλο των διαχειριστών τους τα τελευταία χρόνια. Η ευελιξία που παρέχουν οι τεχνολογίες εικονικοποίησης, επιτρέπει την ταχεία και απρόσκοπτη ανάπτυξη λογισμικού σε φυσικούς υπολογιστές (bare metal), παρέχοντας τη δυνατότητα ανάπτυξης πολύπλοκων υποδομών πληροφορικής, καθώς επίσης και τη δυνατότητα δοκιμών και πειραματισμού με διάφορες τεχνολογίες πληροφοριών χωρίς περιορισμούς. Αυτό συμβαίνει, διότι μια εικονική μηχανή ή/και συσκευή δύναται να δημιουργείται, να διαμορφώνεται και να διαγράφεται κατά βούληση.

Όπως θα αναφερθεί και θα αναλυθεί εκτενώς στο επόμενο Κεφάλαιο της παρούσας διπλωματικής εργασίας, οι μελέτες περίπτωσης εφαρμογής διαδικασιών αυτοματοποίησης, εξετάστηκαν σε υποδομή αποτελούμενη από εικονικές μηχανές (Virtual Machines, VMs).

Η επιλογή αυτή στηρίχθηκε στο γεγονός ότι οι περισσότεροι οργανισμοί και επιχειρήσεις παρέχουν στο προσωπικό τους υπηρεσίες πληροφορικής, οι οποίες βασίζονται σε υποδομές που έχουν δημιουργηθεί σε περιβάλλοντα εικονικοποίησης. Η τεχνολογία δικτύωσης που καθορίζεται από λογισμικό (Software Defined Network - SDN) είναι μια προσέγγιση στη διαχείριση δικτύου που επιτρέπει μέσω προγραμματισμού μια δυναμική, αποδοτική ρύθμιση των παραμέτρων του, προκειμένου να αυξηθεί η απόδοση και να βελτιωθεί η παρακολούθηση του, με αποτέλεσμα να προσομοιάζει περισσότερο με υπολογιστική νέφους (cloud computing).

Η τεχνολογία δικτύωσης που καθορίζεται από λογισμικό (SDN) προορίζεται να αντιμετωπίσει την προβληματική της στατικής αρχιτεκτονικής των παραδοσιακών δικτύων, όπου απαντάται ένας αποκεντρωτικός και πολύπλοκος τρόπος λειτουργίας, ενώ τα σύγχρονα δίκτυα απαιτούν

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

μεγαλύτερη ευελιξία και ευκολότερη αντιμετώπιση και επίλυση των όποιων εμφανιζόμενων προβλημάτων. Επιπλέον, μια εικονική μηχανή (virtual machine), θα πρέπει να είναι εφικτό να εγκατασταθεί σε οποιαδήποτε υποδομή υπολογιστικού νέφους (cloud infrastructure), είτε ιδιωτικού (private cloud), είτε δημόσιου (public), είτε υβριδικού (hybrid cloud), χωρίς καμία διαφοροποίηση. Το μόνο σημείο, που πρέπει να ληφθεί υπόψιν είναι τα μεγέθη χωρητικότητας των εικονικών μηχανών (VMs) [αριθμός επεξεργαστών (CPUs), χωρητικότητα μνήμης «RAM», χωρητικότητα δίσκου] να είναι μεγαλύτερα των εκάστοτε ελάχιστα απαιτούμενων. Επίσης, το λειτουργικό σύστημα θα πρέπει να είναι συμβατό με τον εκάστοτε επιλεχθέντα τρόπο υλοποίησης.

Για την υλοποίηση των μελετών περίπτωσης αυτοματοποίησης εργασιών και διαδικασιών πληροφορικής, επιλέχθηκε να χρησιμοποιηθεί κάποιο σύστημα διαχείρισης διαμόρφωσης λογισμικού (Software Configuration Management - SCM). Η πρακτική του εν λόγω συστήματος «SCM» αναφέρεται στην καταγραφή των αλλαγών [παραμετροποιήσεις (configurations), εγκατάσταση πακέτων λογισμικού κ.ά.], που πραγματοποιούνται σε έναν εξυπηρετητή (server) (είτε αυτός είναι φυσικός, είτε εικονική μηχανή), είτε στους Υπολογιστές – Πελάτες του δικτύου (Clients/Hosts). Κάποια από τα ευρέως χρησιμοποιούμενα λογισμικά διαχείρισης διαμόρφωσης αναλύονται στη συνέχεια⁷.

2.2 Τάσεις και Μέρη Αγοράς

Η συνολική αγορά λογισμικού διαχείρισης αλλαγών και διαμόρφωσης αναμένεται να αυξηθεί από 1,29 δισ. \$ το 2016 σε 2,04 δισ. \$ έως το 2021, με Σύνθετο Ετήσιο Ρυθμό Ανάπτυξης (Compound Annual Growth Rate - CAGR) 11,0% από το 2016 έως το 2021. Τα οφέλη της χρήσης λογισμικών διαχείρισης διαμόρφωσης (configuration management software) περιλαμβάνουν τη δημιουργία ενός αξιόπιστου αποθετηρίου (repository) με ακριβείς πληροφορίες σχετικά με την εκάστοτε υποδομή πληροφορικής. Οι λύσεις διαχείρισης διαμόρφωσης στοχεύουν στην παροχή αξιόπιστων πληροφοριών διαμόρφωσης, οι οποίες θα υποστηρίξουν τη λήψη αποφάσεων και

⁷ https://en.wikipedia.org/wiki/Software-defined_networking

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

θα συνεισφέρουν στην ταχύτερη επίλυση συμβάντων και προβλημάτων, τα οποία, με τη σειρά τους, θα υποβοηθήσουν την ανάπτυξη της αγοράς. Η αγορά λογισμικών διαχείρισης αλλαγών και διαμόρφωσης επεκτείνεται, καθώς οι οργανισμοί εφαρμόζουν τέτοιου είδους λύσεις για τη θέσπιση αποτελεσματικών κανόνων που μπορούν να μειώσουν τους χρόνους διακοπής λειτουργίας του συστήματος, να βελτιώσουν τη συμμόρφωση με τις οδηγίες και τους κανόνες του εκάστοτε οργανισμού, να αυτοματοποιήσουν ολόκληρη την υποδομή σε κάποιο βαθμό και να βελτιώσουν την ευελιξία της, επιτρέποντάς τους να επικεντρώνονται στις βασικές επιχειρηματικές τους λειτουργίες⁸.

Τα τελευταία χρόνια, η μέθοδος «DevOps» [Software Development (Dev) and IT Operations (Ops) - Ανάπτυξη λογισμικού και Λειτουργίες Τεχνολογίας Πληροφοριών] έχει μεταλλαχθεί από μια δευτερεύουσα, σε μια κύρια οντότητα που πρέπει να υφίσταται σε οποιαδήποτε επιχείρηση διαθέτει υποδομή πληροφορικής. Εργασίες όπως, η παροχή περιβάλλοντος λειτουργίας, η ανάπτυξη εφαρμογών και η συντήρηση υποδομών, εκτελούνται παραδοσιακά με μη αυτοματοποιημένο τρόπο. Προκειμένου να εξαλειφθεί το στοιχείο του ανθρώπινου σφάλματος που σχετίζεται με αυτές τις εργασίες, αναπτύχθηκαν τα εργαλεία διαχείρισης διαμόρφωσης (Configuration Management Tools).

Οι συνθήκες του τεχνολογικού περιβάλλοντος, όπου βρίσκουν εφαρμογή τα εργαλεία διαχείρισης διαμόρφωσης, έχουν ωριμάσει σημαντικά τα τελευταία χρόνια και η αγορά αναμένεται να γνωρίσει περαιτέρω εξέλιξη και στο μέλλον. Επομένως, οι εξελίξεις αναφορικά με το χώρο της διαχείρισης διαμόρφωσης θα παραμείνουν ως μια από τις σημαντικότερες ανησυχίες όλων των ομάδων πληροφορικής που «ασπάζονται» τη μέθοδο «DevOps» στο άμεσο μέλλον.

Ουσιώδεις παράγοντες που διαδραματίζουν βασικό ρόλο στην αυξανόμενη δημοτικότητα των εργαλείων διαχείρισης διαμόρφωσης είναι οι δυνατότητές που διαθέτουν για αποθήκευση πληροφοριών και εξαρτώνται από την εκάστοτε έκδοση του λογισμικού, η απαίτηση για μείωση της πολυπλοκότητας του συστήματος, η εμφάνιση της έννοιας του υπολογιστικού νέφους

⁸<https://www.marketsandmarkets.com/Market-Reports/change-configuration-management-market-103332706.html>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

(cloud), καθώς και η αυξανόμενη ανάγκη για λεπτομερή εικόνα επί του διαθέσιμου υλικού και λογισμικού ενός οργανισμού ή μιας επιχείρησης.

Επιπλέον, ο αριθμός επιχειρήσεων που συνειδητοποιούν ότι πλέον δεν μπορούν να βασίζονται σε ένα συνονθύλευμα συστημάτων ή σε «ad-hoc» συστήματα διαχείρισης της διαμόρφωσης, ολοένα και αυξάνεται. Κατ' επέκταση αυξάνεται και η προτίμησή τους για εργαλεία διαχείρισης διαμόρφωσης, προτίμηση η οποία αναμένεται να προωθήσει περαιτέρω την ανάπτυξη της εν λόγω αγοράς.

Η παγκόσμια αγορά εργαλείων διαχείρισης διαμόρφωσης μπορεί να καταταμηθεί με βάση τον τρόπο ανάπτυξης, τον τύπο της παρεχόμενης λύσης, και τον τελικό χρήστη. Με βάση τον τρόπο ανάπτυξης, η αγορά μπορεί να ταξινομηθεί σε εργαλεία που θα αναπτυχθούν στις εγκαταστάσεις (on premise) του οργανισμού, της επιχείρησης κλπ, καθώς και σε αυτά μέσω υπολογιστικού νέφους (cloud). Με βάση τον τύπο προϊόντος, στη χρήση τεχνολογιών αυτοματοποίησης και στον αντίποδα στη χρήση τεχνολογιών Διαχείρισης Διαμόρφωσης Βάσεων Δεδομένων (Configuration Management Database - CMDB). Με βάση τον τελικό χρήστη, η αγορά μπορεί να ταξινομηθεί ως Τεχνολογίας Πληροφοριών και Τηλεπικοινωνιών (IT & Telecom), στρατιωτική, πολιτική υποδομή, και άλλα.

Οι σημαντικότεροι «παίκτες» που δραστηριοποιούνται στην αγορά εργαλείων διαχείρισης διαμόρφωσης είναι οι εταιρείες «Chef», «Puppet», «CFEngine AS», «Red Hat Inc.» (Ansible), «SaltStack Inc.», «Canonical Ltd.» (Juju), «HashiCorp» (Vagrant) και «Microsoft» (PowerShell DSC).

Στην *Εικόνα 1* διακρίνονται μερικά σημαντικά εργαλεία διαχείρισης διαμόρφωσης της αγοράς, κάποια εκ των οποίων θα αναλυθούν στη συνέχεια. Φυσικά υπάρχουν και περισσότερα εργαλεία ανάλογα με τις ανάγκες και τις απαιτήσεις του συγκεκριμένου οργανισμού ή της εκάστοτε επιχείρησης. Λίγα εξ αυτών μπορούν να λειτουργήσουν συμπληρωματικά έχοντας καλή συνολική απόδοση, ενώ κάποια άλλα απλά υπερέχουν σε συγκεκριμένες δραστηριότητες.

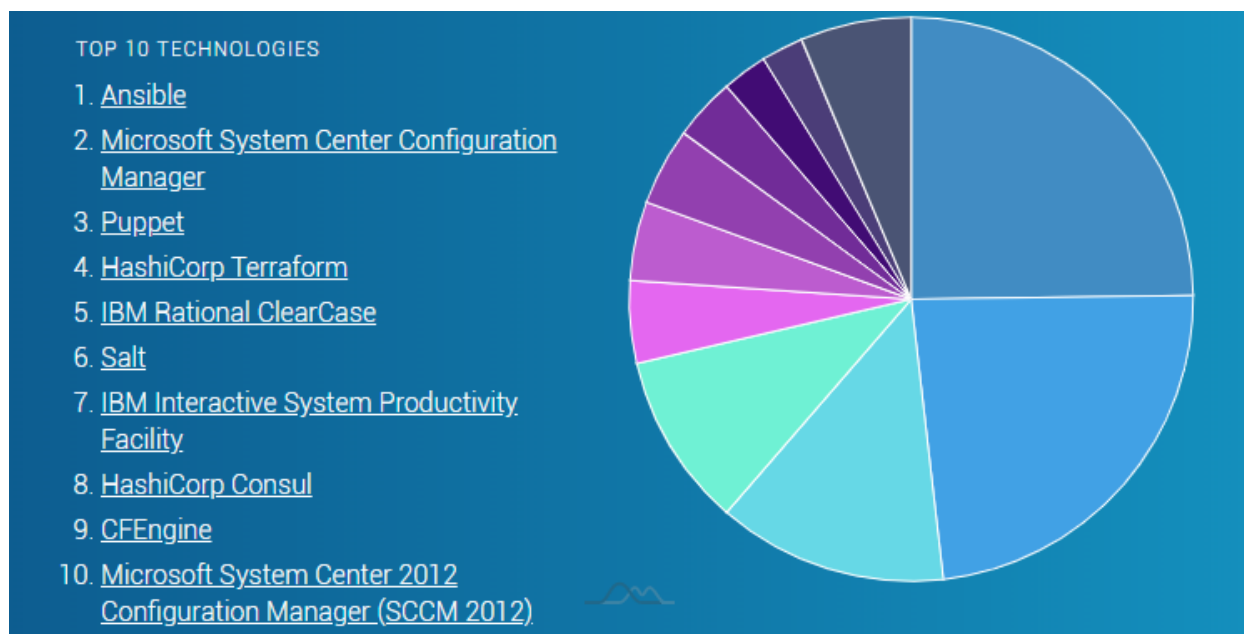
“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Στην *Εικόνα 1* και στον *Πίνακα 1* εμφανίζονται τα εργαλεία διαμόρφωσης που καταλαμβάνουν το σημαντικότερα ποσοστά μεριδίων της αγοράς κατά το 2020 παγκοσμίως⁹.

Πίνακας 1: Μερίδια αγοράς λογισμικών διαχείρισης διαμόρφωσης σε παγκόσμιο επίπεδο.

Κατάταξη	Τεχνολογία	Μερίδιο Αγοράς
1	Ansible	24.79%
2	Microsoft System Center Configuration Manager	23.39%
3	Puppet	13.17%
4	HashiCorp Terraform	9.99%
5	IBM Rational ClearCase	4.72%
6	Salt	4.53%
7	IBM Interactive System Productivity Facility	4.45%
8	HashiCorp Consul	3.59%
9	CFEngine	2.61%
10	Microsoft System Center 2012 Configuration Manager (SCCM 2012)	2.40%



Εικόνα 1: Σειρά κατάταξης σε παγκόσμιο επίπεδο των λογισμικών διαχείρισης διαμόρφωσης.

⁹<https://www.datanyze.com/market-share/configuration-management-313/Datanyze%20Universe?page=1>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

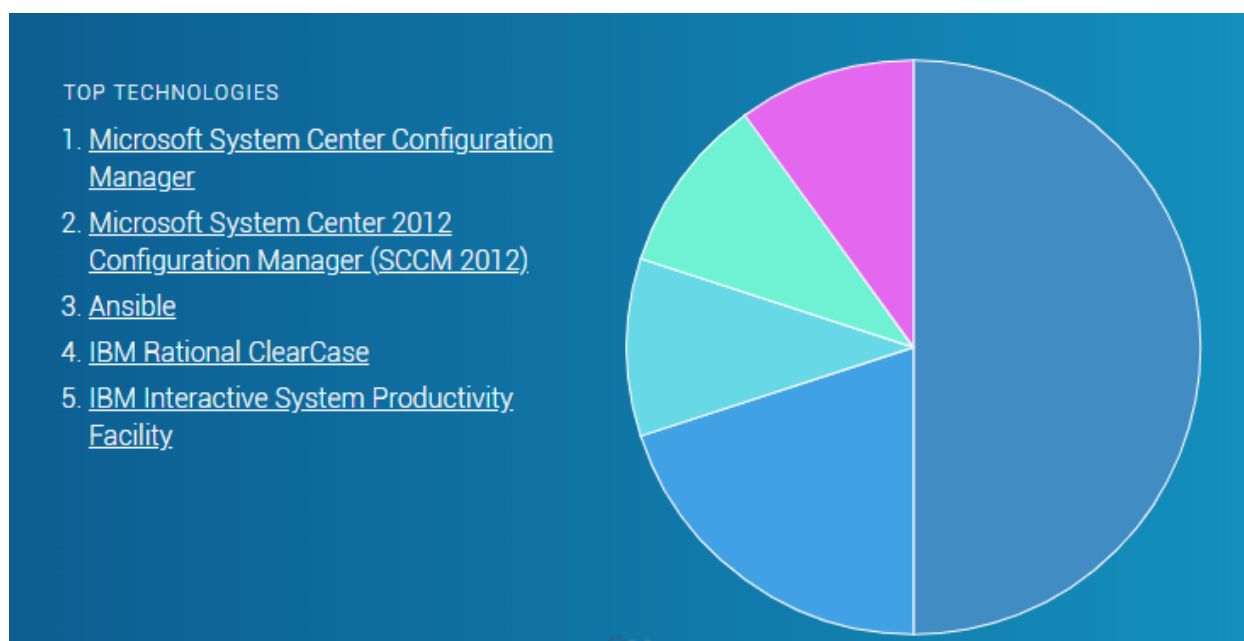
Ελισάβετ Αραμπατζή

Στην *Εικόνα 2* και στον *Πίνακα 2* παρουσιάζονται τα μερίδια αγοράς στη χώρα μας κατά το 2020¹⁰.

¹⁰ <https://www.datanyze.com/market-share/configuration-management--313/Greece>

Πίνακας 2: Μέρη αγοράς λογισμικών διαχείρισης διαμόρφωσης στην Ελλάδα.

Κατάταξη	Τεχνολογία	Μερίδιο Αγοράς
1	Microsoft System Center Configuration Manager	50.00%
2	Microsoft System Center 2012 Configuration Manager (SCCM 2012)	20.00%
3	Ansible	10.00%
4	IBM Rational ClearCase	10.00%
5	IBM Interactive System Productivity Facility	10.00%



Εικόνα 2: Σειρά κατάταξης των λογισμικών διαχείρισης διαμόρφωσης στην Ελλάδα.

2.3 Εργαλεία Αυτοματοποίησης Διαδικασιών Τεχνολογίας Πληροφοριών

2.3.1 Εργαλείο «CFEngine»

Το εργαλείο «CFEngine» αποτελεί ένα περιβάλλον λογισμικού (framework) αυτοματοποίησης υποδομής πληροφορικής και συνεχούς λειτουργίας που βοηθά τους μηχανικούς, τους διαχειριστές πληροφοριακών συστημάτων και άλλους ενδιαφερόμενους στο τμήμα Πληροφορικής ενός οργανισμού ή μιας επιχείρησης να διαχειρίζονται την υποδομή

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

πληροφορικής, διασφαλίζοντας παράλληλα το επίπεδο των υπηρεσιών και τη συμμόρφωση με τις πολιτικές του εκάστοτε οργανισμού ή της επιχείρησης¹¹.

Το «CFEngine» χρησιμοποιεί αυτόνομους πράκτορες (agents) που εκτελούνται σε κάθε κόμβο της υποδομής, εφαρμόζοντας την επιθυμητή κατάσταση και αποστέλλοντας συνεχώς αναφορές στους διαχειριστές των πληροφοριακών συστημάτων, σχετικά με το εάν εμφανίσθηκαν τυχόν προβλήματα, καθώς και εάν ολοκληρωθήκαν επιτυχώς τα όσα είχαν προγραμματισθεί προς υλοποίηση¹². Το «CFEngine» εκτελείται στις μικρότερες ενσωματωμένες συσκευές του συστήματος, σε εξυπηρετητές, στο υπολογιστικό νέφος (cloud) και σε κεντρικούς υπολογιστές (mainframes), διαχειρίζοντας εύκολα δεκάδες χιλιάδες από αυτούς¹³.

Αναπτύχθηκε από τον Mark Burgess το 1993 και είναι διαθέσιμο τόσο ως λογισμικό ανοικτού κώδικα όσο και ως εμπορικό λογισμικό. Η ανοικτού κώδικα έκδοση υποστηρίζει έως και είκοσι πέντε (25) κόμβους (nodes), ενώ από εκεί και έπειτα η τιμή εξαρτάται από το μέγεθος της υποδομής. Υποστηρίζει λειτουργικά συστήματα «UNIX», «Windows», καθώς και μεικτά (cross – platform) περιβάλλοντα και χρησιμοποιείται από μεγάλες εταιρείες όπως «Intel», «AT&T», «LinkedIn», «Amazon», «State Farm», «SalesForce» κ.λπ.¹⁴

Στα πλεονεκτήματά του συγκαταλέγονται η υψηλή διαθεσιμότητα, η υψηλή επεκτασιμότητα, το υψηλό επίπεδο ασφάλειας που παρέχει, η ταχύτητά του, καθώς και η χαμηλή κατανάλωση σε πόρους («CPU», «RAM Memory» κλπ)¹⁵.

Από την άλλη πλευρά ωστόσο, η τεκμηρίωση (documentation) που παρέχεται είναι εξαιρετικά δυσνόητη με αποτέλεσμα η αρχική εγκατάσταση να παρουσιάζει αυξημένο βαθμό δυσκολίας. Επιπλέον, αναφέρεται ότι η παραμετροποίησή του είναι αρκετά πολύπλοκη και δεν είναι τόσο αποδοτικό για τον έλεγχο της ακεραιότητας κρίσιμων αρχείων¹⁶.

¹¹<https://cfengine.com/product/>·<https://en.wikipedia.org/wiki/CFEngine>·<https://www.upguard.com/articles/configuration-management-tools>

¹² <https://northern.tech/products/cfengine>· <https://cfengine.com/product/>

¹³ <https://cfengine.com/product/what-is-cfengine/>

¹⁴ <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

¹⁵ <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

¹⁶ <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

Το «CFEngine» μπορεί να χρησιμοποιηθεί για την αυτοματοποίηση κάθε είδους υποδομής πληροφορικής. Ένα παράδειγμα αποτελούν οι εξυπηρετητές (servers). Οι εξυπηρετητές (servers) χρειάζονται συνεχείς, επαναλαμβανόμενες διαμορφώσεις, οι οποίες πρέπει να τίθενται υπό παρακολούθηση για πολλούς λόγους. Αρχικά, πρέπει να παρέχουν ένα περιβάλλον όπου θα είναι γνωστό πως τα προγράμματα εκτελούνται σωστά, θα πρέπει να εντοπίζουν τυχόν προβλήματα συγκρίνοντας την τρέχουσα κατάσταση με μια γνωστή γραμμή βάσης (baseline), για να διασφαλιστεί η ασφάλεια σε κάθε σύστημα και ούτω καθεξής. Κάθε φορά, όμως, που κάποιος τροποποιεί τη διαμόρφωση ενός μηχανήματος με μη αυτοματοποιημένο τρόπο, η δυνατότητα πρόβλεψης της όποιας κατάστασης μειώνεται, λόγω σφαλμάτων που οφείλονται στη χειροκίνητη διαμόρφωση ή παραλλαγών. Με την πάροδο του χρόνου η διαμόρφωση ενός μεγάλου αριθμού μηχανημάτων τείνει να διαφοροποιείται αρκετά, μετατρέποντας της διαχείριση τους σε ένα εξαιρετικό δύσκολο εγχείρημα¹⁷.

Το «CFEngine», παρέχει ενδεικτικά τις κάτωθι δυνατότητες¹⁸:

➤ **Διαμόρφωση.**

Η διαμόρφωση τόσο του βασικού λειτουργικού συστήματος, όσο και του εγκατεστημένου λογισμικού μπορεί εύκολα να υλοποιηθεί χρησιμοποιώντας το «CFEngine», διατηρώντας τα ενημερωμένα.

➤ **Διαχείριση χρηστών.**

Το «CFEngine» επιτρέπει τον έλεγχο των λογαριασμών των χρηστών και τα χαρακτηριστικά αυτών, δίνει τη δυνατότητα υπόδειξης των λογαριασμών χρηστών που χρειάζονται και επιπλέον, υποστηρίζει τον έλεγχο συγκεκριμένων παραμέτρων όπως κωδικοί πρόσβασης, ημερομηνίες λήξης λογαριασμών κ.λπ.

➤ **Εγκατάσταση λογισμικού.**

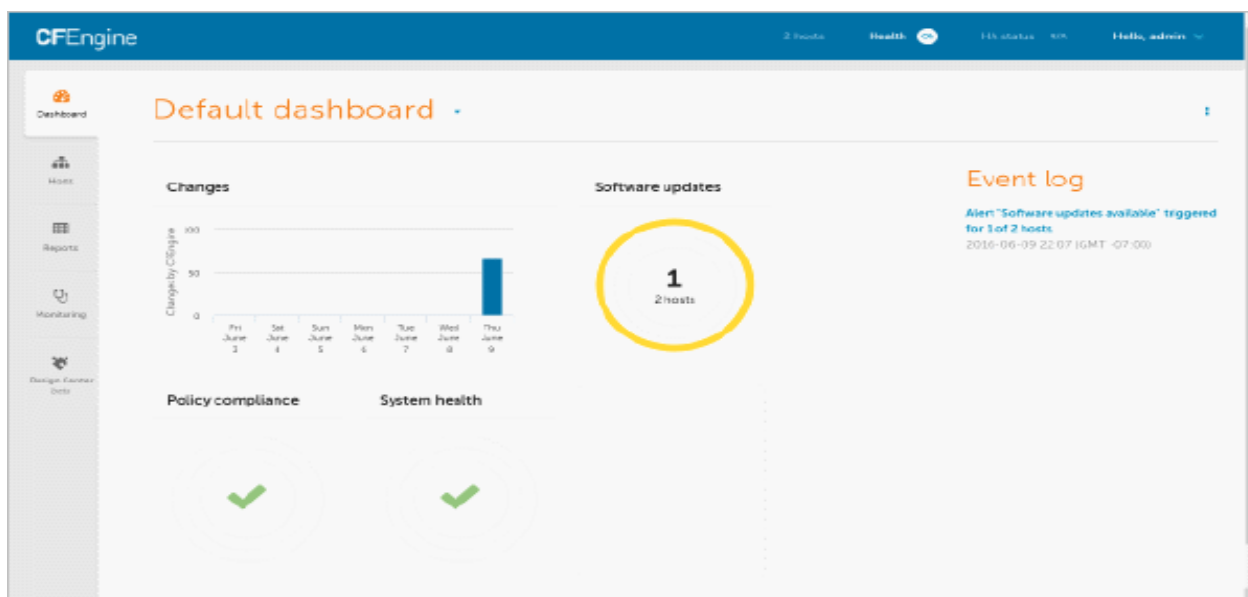
¹⁷ <https://www.oreilly.com/library/view/learning-cfengine-3/9781449334536/ch01.html>

¹⁸ <https://www.oreilly.com/library/view/learning-cfengine-3/9781449334536/ch01.html>

Το «CFEngine» παρέχει τη δυνατότητα διαχείρισης και παραμετροποίησης λογισμικού, συμπεριλαμβανομένων εγκαταστάσεων, αναβαθμίσεων και καταργήσεων. Το «CFEngine» έχει σχεδιαστεί για να αλληλεπιδρά με τα εγγενή (native) εργαλεία διαχείρισης πακέτων λογισμικού του συστήματος, έτσι ώστε η διαχείριση του λογισμικού να διενεργείται με τον καταλληλότερο τρόπο. Το «CFEngine» μπορεί, επίσης, να χρησιμοποιηθεί για τη μη αυτοματοποιημένη εγκατάσταση ή κατάργηση λογισμικού.

➤ **Ασφάλεια και συμμόρφωση.**

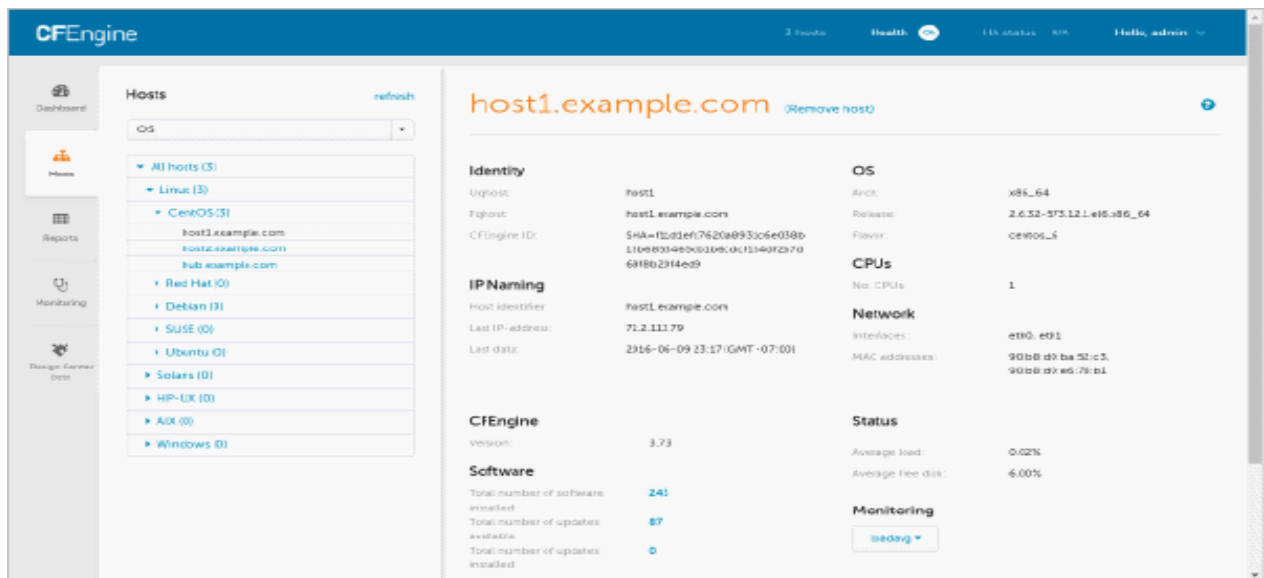
Η έννοια της ασφάλειας περιλαμβάνει πολλές πτυχές ενός συστήματος, όπως δικαιώματα πρόσβασης σε αρχεία, δικαιώματα χρήστη, διαμόρφωση και κατάσταση υπηρεσιών (services) και πολλές άλλες. Οι πτυχές αυτές καθίστανται εύκολα διαχειρίσιμες με το «CFEngine», ενσωματώνοντας σε έναν εξυπηρετητή (server) πολιτικές ασφαλείας.



Εικόνα 3: Απεικόνιση διεπαφής ιστού (web interface) διαχειριστή CFEngine.

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή



Εικόνα 4: Απεικόνιση διεπαφής ιστού (web interface) διαχειριστή CFEngine ως προς ένα συγκεκριμένο κόμβο.

2.3.2 Εργαλείο «Rudder»

Το εργαλείο «Rudder» είναι ένα βοηθητικό πρόγραμμα ελέγχου και διαμόρφωσης ανοικτού κώδικα που βοηθά στην αυτοματοποίηση της διαμόρφωσης συστήματος σε μεγάλες υποδομές πληροφορικής. Το «Rudder» βασίζεται σε έναν «ελαφρύ», τοπικό «πράκτορα» (light-weight agent), εγκατεστημένο σε κάθε διαχειριζόμενο μηχάνημα, ενώ υποστηρίζει λειτουργικά συστήματα «Unix», «Microsoft Windows», «Android» και «Ubuntu»¹⁹.

Η κοστολόγηση αφορά την περίπτωση επιλογής πραγματοποίησης συνδρομής. Το «Rudder» μπορεί να χρησιμοποιηθεί και χωρίς συνδρομή (subscription), με κάποιους περιορισμούς στην παροχή υπηρεσιών, δυνατοτήτων (features) και κάποιων πρόσθετων λειτουργιών (plugins), καθώς επίσης και στην υποστήριξη και συντήρηση της υποδομής του «Rudder». Στην επίσημη ιστοσελίδα δεν αναφέρεται ακριβής κοστολόγηση, αλλά μόνο ότι η συνδρομή παρέχεται σε τουλάχιστον ετήσια βάση και ανά μηχάνημα (annual per machine subscription)²⁰.

¹⁹ [https://en.wikipedia.org/wiki/Rudder_\(software\)](https://en.wikipedia.org/wiki/Rudder_(software))·<https://www.upguard.com/articles/configuration-management-tools>

²⁰ <https://www.rudder.io/pricing/subscription/>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Το «Rudder» παράγεται από την εταιρεία «Normation» και ιδρύθηκε το 2010. Η διεπαφή ιστού (web interface) από την πλευρά του εξυπηρετητή (server) έχει αναπτυχθεί στη γλώσσα προγραμματισμού «Scala» και ο τοπικός «πράκτορας» (agent) στη γλώσσα «C» και δημοσιεύονται ως ελεύθερο λογισμικό σύμφωνα με το «General Public License» (GNU) έκδοση 3.0²¹.

Το εργαλείο «Rudder» παρέχει μία διεπαφή ιστού (web interface) για τη διαχείριση των κόμβων και τον καθορισμό πολιτικών. Επιπλέον, φιλοξενεί το αρχείο «inventory» και παρέχει έναν προσαρμοσμένο επεξεργαστή πολιτικών, ο οποίος είναι μοναδικός. Το «Rudder», επίσης, αυτοματοποιεί τις απλές εργασίες διαχείρισης όπως την εγκατάσταση ή τη διαμόρφωση πακέτων λογισμικού. Υποστηρίζει τη δυνατότητα «FULL REST API» για την επικοινωνία του «Rudder Agent» με τον «Rudder Server» και διαθέτει το σύστημα ελέγχου εκδόσεων «GIT» στο «backend» του. Τέλος, δύναται να δημιουργεί δυναμικά κάθε πολιτική που αφορά κάθε φυσικό ή ηλεκτρονικό υπολογιστή - πελάτη (host)²².

Στα πλεονεκτήματά του συγκαταλέγεται ότι το «Rudder» βασίζεται στο πρότυπο του «CFEngine» και έτσι κληρονομεί κάποιες λειτουργίες του εργαλείου «CFEngine», που αναφέρθηκαν παραπάνω. Επιπλέον, παρέχει αυτοματοποιημένο αρχείο «inventory» τόσο για υλικό, όσο και για λογισμικό, παρέχει αναφορές με γραφική αναπαράσταση και περιλαμβάνει τη βιβλιοθήκη βέλτιστων πρακτικών χρήσης του εργαλείου²³.

Σε ό,τι αφορά τα μειονεκτήματα του εν λόγω εργαλείου, η κοινότητα του «Rudder» αναπτύσσεται αλλά δεν είναι πολύ μεγάλη σήμερα, όπως η κοινότητα των εργαλείων «Puppet» και «Ansible» που θα περιγραφούν στη συνέχεια, με αποτέλεσμα να μην είναι ιδιαίτερα εύκολη η πρόσβαση σε διάφορες δυνατότητες ανταλλαγής απόψεων, υποστήριξης πρακτικών και επίλυσης ανακυπτόντων ζητημάτων²⁴.

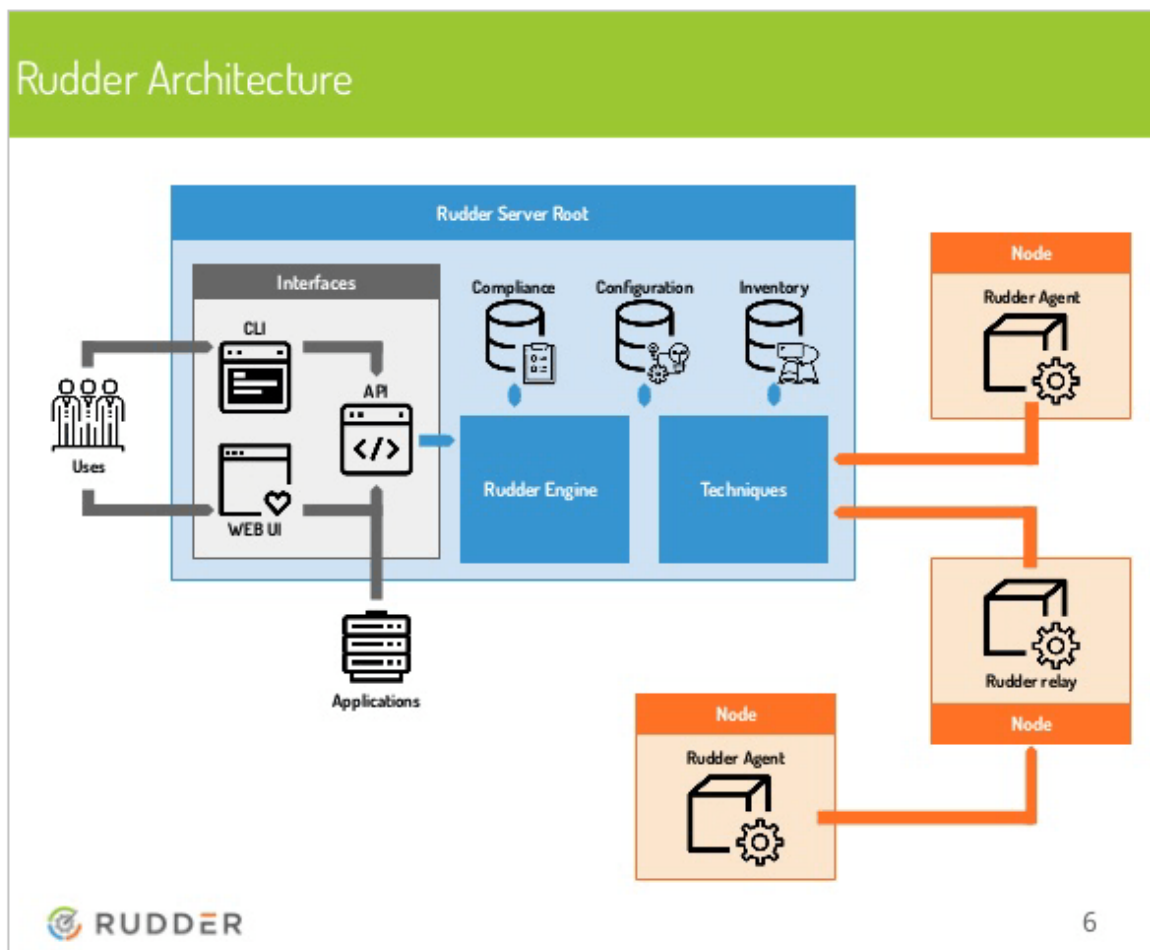
²¹ <https://docs.rudder.io/reference/6.0/index.html>

²² <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

²³ <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

²⁴ <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

Επισημαίνεται ότι, η χρήση του εργαλείου «Rudder» για την αυτοματοποίηση ενεργειών που απαιτούνται να εκτελεστούν μόνο μια φορά, μπορεί να χαρακτηριστεί υπερβολική.



Εικόνα 5: Αρχιτεκτονική του εργαλείου «Rudder».

2.3.3 Εργαλείο «Puppet»

Το εργαλείο «Puppet» είναι ένα εργαλείο διαχείρισης διαμόρφωσης λογισμικού ανοικτού κώδικα. Χρησιμοποιείται για την ανάπτυξη, διαμόρφωση και διαχείριση εξυπηρετητών (servers). Χρησιμοποιεί αρχιτεκτονική «master – slave». Οι διαμορφώσεις αντλούνται από τον κύριο (master) προς τους κόμβους (nodes). Υπάρχουν δύο εκδόσεις του «Puppet», το «Puppet Enterprise» και το «Open Source Puppet». Εκτός από την παροχή λειτουργιών του «Open Source Puppet», το «Puppet Enterprise» παρέχει, επίσης, εργαλεία γραφικού περιβάλλοντος χρήστη «GUI» (Graphical User Interface), περιβάλλον προγραμματισμού εφαρμογών «API» (Application

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Programming Interface) και γραμμής εντολών (Command Line Interface - CLI) για τη διαχείριση κόμβων. Το «Puppet Enterprise» παρέχεται δωρεάν για έως και δέκα (10) κόμβους²⁵.

Το λογισμικό αυτοματοποίησης «Puppet» χρησιμοποιεί τη δηλωτική γλώσσα της «Puppet» για τη διαχείριση διαφόρων σταδίων της υποδομής πληροφορικής, συμπεριλαμβανομένης της παροχής, επιδιόρθωσης, διαμόρφωσης και διαχείρισης λειτουργικών συστημάτων και στοιχείων εφαρμογών σε κέντρα δεδομένων επιχειρήσεων ή οργανισμών και υποδομές υπολογιστικού νέφους (cloud). Είναι συμβατό με πολλά λειτουργικά συστήματα που προσομοιάζουν με «Unix», καθώς και με «Microsoft Windows», και περιλαμβάνει τη δική του δηλωτική γλώσσα για να περιγράψει τη διαμόρφωση του συστήματος. Έχει αναπτυχθεί σε γλώσσα «C++», «Clojure» και «Ruby», με την έκδοση ελεύθερου λογισμικού που κυκλοφόρησε με την άδεια «Apache License 2.0» (GNU - General Public License (GPL) έως την έκδοση 2.7.0). Κάποιοι από τους οργανισμούς που αξιοποιούν το «Puppet» και τις δυνατότητές του είναι οι «JP Morgan Chase», «OnxyPoint», «CBSButler», «Heart Land», «AT&T», «Smart School» κ.λπ.

Ο χρήστης περιγράφει τους πόρους του συστήματος και την κατάστασή του, είτε χρησιμοποιώντας τη δηλωτική γλώσσα του «Puppet», είτε τη γλώσσα «Ruby DSL» (γλώσσα για συγκεκριμένο τομέα – Domain Specific Language). Αυτές οι πληροφορίες αποθηκεύονται σε αρχεία με την ονομασία «Puppet manifests». Το «Puppet» ανακαλύπτει τις πληροφορίες συστήματος μέσω ενός βοηθητικού προγράμματος που ονομάζεται «Facter» και μεταγλωττίζει τις εκδηλώσεις «Puppet» σε έναν κατάλογο για συγκεκριμένο σύστημα που περιέχει πόρους και εξάρτηση πόρων, οι οποίοι εφαρμόζονται έναντι των συστημάτων που απαιτούνται.

Το «Puppet» αποτελείται από μια προσαρμοσμένη δηλωτική γλώσσα, που έχει σκοπό να περιγράψει τη διαμόρφωση του συστήματος, η οποία είτε μπορεί να εφαρμοστεί απευθείας στο σύστημα, είτε να μεταγλωττιστεί σε έναν κατάλογο και να διανεμηθεί στο σύστημα προορισμού μέσω παραδείγματος χάρη του μοντέλου πελάτη-εξυπηρετητή (client – server). Αυτό επιτυγχάνεται χρησιμοποιώντας μία διεπαφή τύπου «REST API» (Representational State Transfer Application Programming Interface) και ο πράκτορας (agent) χρησιμοποιεί σύστημα για

²⁵[https://en.wikipedia.org/wiki/Puppet_\(company\)](https://en.wikipedia.org/wiki/Puppet_(company))·<https://www.upguard.com/articles/configuration-management-tools>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

την επιβολή του πόρου που καθορίζεται στις δηλώσεις. Το επίπεδο αφαίρεσης πόρων επιτρέπει στους διαχειριστές πληροφοριακών συστημάτων να περιγράψουν τη διαμόρφωση με όρους υψηλού επιπέδου, όπως χρήστες, υπηρεσίες και πακέτα χωρίς την ανάγκη καθορισμού συγκεκριμένων εντολών λειτουργικού συστήματος (όπως «rpm», «yum», «apt»). Το «Puppet» χαρακτηρίζεται ως σχετικά εύκολο εργαλείο προς χρήση αφού απαιτεί περιορισμένες γνώσεις προγραμματισμού.

Η αρχιτεκτονική του «Puppet» ακολουθεί συνήθως το μοντέλο πελάτη-εξυπηρετητή (client – server). Ο πελάτης είναι γνωστός ως «agent» και ο εξυπηρετητής είναι γνωστός ως «master». Για δοκιμή και απλή διαμόρφωση, μπορεί επίσης να χρησιμοποιηθεί και ως αυτόνομη (standalone) εφαρμογή που εκτελείται από τη γραμμή εντολών²⁶.

Ο «Puppet Server» είναι εγκατεστημένος σε έναν ή περισσότερους εξυπηρετητές (servers) και ο «Puppet Agent» είναι εγκατεστημένος σε όλα τα μηχανήματα που ο χρήστης θέλει να διαχειριστεί. Οι «Puppet Agents» επικοινωνούν με τον εξυπηρετητή και λαμβάνουν οδηγίες διαμόρφωσης. Στη συνέχεια, ο «agent» εφαρμόζει την εκάστοτε διαμόρφωση στο σύστημα και στέλνει μια αναφορά κατάστασης στον εξυπηρετητή. Οι συσκευές μπορούν να εκτελούν τον «Puppet Agent» ως υπηρεσία («daemon» για λειτουργικά συστήματα «Linux» ή «service» για λειτουργικά συστήματα «Windows»), που μπορεί να ενεργοποιείται περιοδικά αυτοματοποιημένα ή μπορεί να εκτελείται με μη αυτοματοποιημένο τρόπο (χειροκίνητα) όποτε χρειάζεται.

²⁶ <https://puppet.com/use-cases/windows-infrastructure-automation/>

Interface). Επιπλέον, κατά την εγκατάστασή του το «Puppet» στερείται επαρκούς ανταλλαγής μηνυμάτων σφάλματος προς τον διαχειριστή μιας υποδομής πληροφορικής²⁸.

2.3.4 Εργαλείο «Chef»

Η «Chef» είναι μια εταιρεία, η οποία έχει αναπτύξει το ομώνυμο εργαλείο διαχείρισης διαμόρφωσης σε γλώσσα προγραμματισμού «Ruby» (πελάτης - client) και «Erlang» (εξυπηρετητής - server). Χρησιμοποιεί τη «Ruby», γλώσσα για συγκεκριμένο τομέα (Domain Specific Language - DSL), για τη σύνταξη των διαμορφώσεων συστήματος οι αποκαλούμενες «συνταγές» (recipes). Το «Chef» χρησιμοποιείται για τον εξορθολογισμό εργασιών διαμόρφωσης και συντήρησης εξυπηρετητών ενός οργανισμού ή μιας επιχείρησης και μπορεί να ενσωματωθεί σε πλατφόρμες που βασίζονται στην υπολογιστική νέφους (cloud) όπως το «Amazon EC2», το «Google Cloud Platform», το «Oracle Cloud», το «OpenStack», το «SoftLayer», το «Microsoft Azure» και το «Rackspace» για την αυτόματη παροχή και διαμόρφωση νέων μηχανημάτων. Μπορεί, επίσης, να χρησιμοποιηθεί και σε υποδομές πληροφοριακών συστημάτων, που έχουν αναπτυχθεί στις εγκαταστάσεις ενός οργανισμού ή μιας επιχείρησης (on – premise) και παρέχει λύσεις για συστήματα μικρής και μεγάλης κλίμακας²⁹.

Στην παροχή του ως λογισμικού ανοιχτού κώδικα (open source), παρέχεται εντελώς δωρεάν. Το «Hosted Chef» που περιλαμβάνει τη χρήση του εν λόγω εργαλείου σε κάποιον οργανισμό, ο οποίος διαθέτει υποδομή με τη χρήση υπολογιστικού νέφους (cloud), παρέχεται σε διάφορες τιμές ανάλογα με τις δυνατότητες που προσφέρει³⁰.

Σε ό,τι αφορά τη χρήση του «Chef» ο χρήστης συντάσσει «συνταγές» (recipes), που περιγράφουν τον τρόπο με τον οποίο το «Chef» διαχειρίζεται εφαρμογές και βοηθητικά προγράμματα εξυπηρετητή (όπως «Apache HTTP Server», «MySQL» ή «Hadoop») και τον τρόπο που αυτά πρέπει να ρυθμιστούν. Αυτές οι «συνταγές», οι οποίες μπορούν να ομαδοποιηθούν

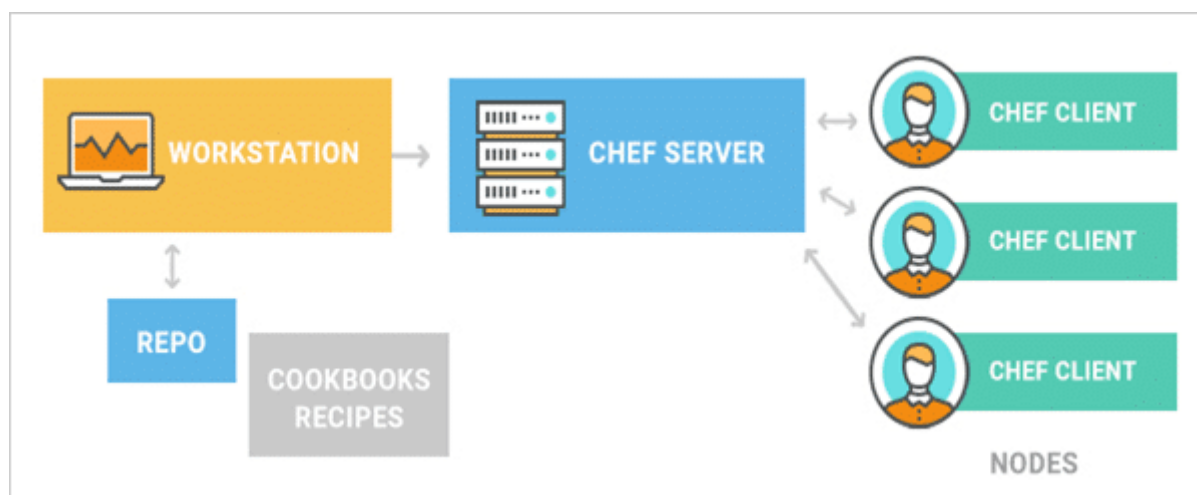
²⁸ <https://www.upguard.com/articles/configuration-management-tools>

²⁹ [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software))· <https://www.upguard.com/articles/configuration-management-tools>

³⁰ <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>· <https://www.chef.io/pricing/>

ως «βιβλίο μαγειρικής» (cookbook) για ευκολότερη διαχείριση, περιγράφουν μια σειρά πόρων που πρέπει να βρίσκονται σε μια συγκεκριμένη κατάσταση, όπως πακέτα λογισμικού που πρέπει να εγκατασταθούν, υπηρεσίες που πρέπει να εκτελούνται ή αρχεία που πρέπει να εγγραφούν. Αυτοί οι διάφοροι πόροι μπορούν να διαμορφωθούν σε συγκεκριμένες εκδόσεις λογισμικού για εκτέλεση και μπορούν να διασφαλίσουν, ότι το λογισμικό εγκαθίσταται με τη σωστή σειρά, με βάση τις σχετικές εξαρτήσεις. Το «Chef» διασφαλίζει ότι κάθε πόρος έχει διαμορφωθεί σωστά και διορθώνει τυχόν πόρους που δεν βρίσκονται στην επιθυμητή κατάσταση.

Το «Chef» μπορεί να εκτελεστεί σε μοντέλο λειτουργίας πελάτη - εξυπηρετητή (Client – Server Model) ή σε αυτόνομη (Standalone) διαμόρφωση με το όνομα «chef-solo». Στο μοντέλο λειτουργίας πελάτη - εξυπηρετητή, ο «πελάτης Chef» στέλνει διάφορα χαρακτηριστικά σχετικά με τον κόμβο στον «εξυπηρετητή Chef». Ο εξυπηρετητής χρησιμοποιεί τη μηχανή «Elasticsearch» για ευρετηρίαση και ταξινόμηση αυτών των χαρακτηριστικών και παρέχει μία διεπαφή «API» (Application Programming Interface) για τους «πελάτες» να υποβάλλουν ερώτημα σε αυτές τις πληροφορίες. Οι «συνταγές» του «Chef» μπορούν να υποβάλουν ερωτήματα για τα χαρακτηριστικά και να χρησιμοποιήσουν τα δεδομένα που προκύπτουν για να βοηθήσουν στη διαμόρφωση του κόμβου³¹.



Εικόνα 7: Αρχιτεκτονική του εργαλείου «Chef»

³¹ [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software)), <https://www.chef.io/it-automation/>

Το εργαλείο «Chef» αποτελεί ουσιαστικά μια πλατφόρμα αυτοματισμού, η οποία παρέχει έναν τρόπο διαμόρφωσης και διαχείρισης και αυτοματοποίησης της υποδομής πληροφορικής ενός οργανισμού ή μιας επιχείρησης. Παραδοσιακά, το «Chef» χρησιμοποιούταν για τη διαχείριση περιβάλλοντος «Linux» σε διανομές όπως «Debian», «Fedora», «Ubuntu» κλπ, αλλά οι νεότερες εκδόσεις υποστηρίζουν και περιβάλλοντα «Microsoft Windows».

Αποτελεί ένα από τα σημαντικότερα συστήματα διαχείρισης διαμόρφωσης μαζί με τα «CFEngine», «Ansible» και «Puppet». Αναπτύχθηκε από τον Adam Jacob και κυκλοφόρησε για πρώτη φορά το 2009, με διαθέσιμη τελευταία σταθερή έκδοση την «14.2.0». Σημαντικοί οργανισμοί που χρησιμοποιούν το συγκεκριμένο εργαλείο είναι οι «Bloom Berg», «BONOBOS», «Facebook», «GE», «Hewlett Packard», «Microsoft», «Yahoo», «Target», «Voxel» κ.λπ. Πέρα από ένα εργαλείο διαχείρισης διαμόρφωσης, το «Chef», μαζί με το «Puppet» και το «Ansible», είναι ένα από τα πιο αξιοσημείωτα εργαλεία υποδομής ως κώδικα (IAC – Infrastructure As Code) της βιομηχανίας αυτοματοποίησης πληροφορικής.

Στα πλεονεκτήματά του συγκαταλέγεται το γεγονός ότι παρέχει ενεργή, έξυπνη και ταχύτατα αναπτυσσόμενη κοινοτική υποστήριξη. Επίσης, ακολουθεί το μοντέλο «Push» και επιτρέπει την υιοθέτηση της υπολογιστικής νέφους (cloud). Επιπλέον, συμβάλλει στην παροχή αξιόπιστων υπηρεσιών, στην ανάπτυξη λογισμικού χωρίς σφάλματα, καθώς τα εντοπίζει πριν αυτά εμφανισθούν και βοηθά στη βελτίωση της διαχείρισης κινδύνων. Οι δυνατότητες αυτοματοποίησης του «Chef» μπορούν να μειώσουν τον κίνδυνο διακοπής λειτουργίας της υποδομής πληροφορικής και να βελτιώσουν τη συμμόρφωση με τους κανόνες και τις πολιτικές ασφαλείας ενός οργανισμού³².

Από την άλλη πλευρά, το εργαλείο «Chef» εκτελείται μόνο σε γλώσσα προγραμματισμού «Ruby». Επίσης, μερικές ροές εργασίας στο «Chef» φαίνονται λίγο περίπλοκες, καθώς το μέγεθος των βάσεων κώδικα (code base) αυξάνεται υπερβολικά³³.

³² <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

³³ [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software)), <https://www.chef.io/why-chef/>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

2.3.5 Εργαλείο «Salt Stack»

Το εργαλείο «Salt», το οποίο αναφέρεται στη βιβλιογραφία ως «Salt Stack», είναι λογισμικό ανοικτού κώδικα, βασίζεται στη γλώσσα προγραμματισμού «Python» και χρησιμοποιείται για αυτοματοποίηση διαδικασιών πληροφορικής, απομακρυσμένη εκτέλεση εργασιών και διαχείριση διαμόρφωσης συστημάτων³⁴. Με τη χρήση του «Salt» επιτυγχάνεται ο έλεγχος και η βελτιστοποίηση οποιασδήποτε υποδομής υπολογιστικού νέφους (cloud-native) ή εσωτερικής εγκατάστασης (on premise) σε μεγάλη κλίμακα. Επιπρόσθετα, παρέχει τη δυνατότητα αυτοματισμών επί ζητημάτων ασφάλειας των πληροφοριακών συστημάτων συμβάλλοντας στον εντοπισμό ευπαθειών και στην αδιάλειπτη επιβολή των κανόνων συμμόρφωσης του οργανισμού ή της επιχείρησης. Τέλος, μπορεί να χρησιμοποιηθεί και για διαδικασίες αυτοματισμών δικτύου παρέχοντας, μέσω μιας ενοποιημένης πλατφόρμας, τη δυνατότητα ελέγχου, διαμόρφωσης και ασφάλειας σε ολόκληρο το δίκτυο³⁵.

Μία επιπλέον δυνατότητα του «Salt» είναι η απομακρυσμένη εκτέλεση εργασιών. Η έμπνευση του «Salt» προήλθε από την ανάγκη των διαχειριστών κέντρων δεδομένων, τεράστιας κλίμακας και πολυπλοκότητας, για τη συλλογή δεδομένων και την εκτέλεση εργασιών σε υψηλές ταχύτητες. Ο δημιουργός του «Salt», Thomas S. Hatch, είχε δημιουργήσει στο παρελθόν πολλά προγράμματα παροχής βοήθειας σε ομάδες πληροφορικής για την επίλυση του προβλήματος της διαχείρισης συστημάτων μεγάλης κλίμακας. Κατάφερε να δημιουργήσει το «Salt», καθώς και άλλες λύσεις ανοικτού κώδικα. Ο Hatch αποφάσισε να χρησιμοποιήσει τη βιβλιοθήκη ανταλλαγής μηνυμάτων «ZeroMQ» για να διευκολύνει τις απαιτήσεις για υψηλή ταχύτητα και δημιούργησε το «Salt» χρησιμοποιώντας το «ZeroMQ» σε όλα τα επίπεδα δικτύωσης.

Στα τέλη Μαΐου του 2011 σημειώθηκε πρόοδος αναφορικά με την παράδοση διαχείρισης διαμόρφωσης, που βασίστηκε στον μηχανισμό απομακρυσμένης εκτέλεσης εργασιών του «Salt». Αυτό το σύστημα διαχείρισης διαμόρφωσης αποθηκεύει όλα τα δεδομένα

³⁴ [https://en.wikipedia.org/wiki/Salt_\(software\)](https://en.wikipedia.org/wiki/Salt_(software))

³⁵ <https://www.saltstack.com/>

διαμόρφωσης (κατάσταση) μέσα σε μια εύκολα κατανοητή δομή δεδομένων που αξιοποιεί τη δηλωτική γλώσσα «YAML».

Η έκδοση «Salt 0.14.0» παρουσίασε ένα προηγμένο σύστημα ελέγχου υπολογιστικού νέφους (cloud) που καθιστά τις ιδιωτικές και δημόσιες εικονικές μηχανές υπολογιστικού νέφους (VM cloud) άμεσα διαχειρίσιμες με το «Salt». Η λειτουργία «Salt Cloud» επιτρέπει την παροχή (provisioning) οποιουδήποτε υπολογιστή υβριδικού υπολογιστικού νέφους (cloud), στη συνέχεια την απομακρυσμένη εκτέλεση του «Salt», τη διαχείριση διαμόρφωσης και τις δυνατότητες αυτοματισμού, προβάλλοντας τις καταστάσεις τους ως συμβάντα στα νέα συστήματα υβριδικού υπολογιστικού νέφους (cloud).

Το «Salt Cloud» υποστηρίζει είκοσι πέντε (25) δημόσια και ιδιωτικά συστήματα υπολογιστικού νέφους (cloud), συμπεριλαμβανομένων των «AWS», «Azure», «VMware», «IBM Cloud» και «OpenStack», παρέχοντας μια διεπαφή για το «Salt» προκειμένου να αλληλεπιδρά με τους υπολογιστές που παρέχονται μέσω υπολογιστικού νέφους και τη λειτουργικότητα που αυτό παρέχει όπως «DNS – Domain Name Service», αποθήκευση, εξισορροπητές φορτίου (loader balancers) κ.λπ³⁶.

Το «Salt» έχει σχεδιαστεί για να είναι εξαιρετικά αρθρωτό και εύκολα επεκτάσιμο, προκειμένου να διευκολύνει τη διαμόρφωση, σε διάφορες περιπτώσεις χρήσης σε μια εκάστοτε υποδομή πληροφορικής. Ο σχεδιασμός των ενότητων (modules) του «Salt» πραγματοποιείται στη γλώσσα «Python» και οι ενότητες αυτές χειρίζονται ορισμένες πτυχές των διαθέσιμων συστημάτων του «Salt». Αυτές οι λειτουργικές μονάδες (modules) επιτρέπουν οι αλληλεπιδράσεις εντός του «Salt», να μπορούν να τροποποιούνται ώστε να ανταποκρίνονται στις ανάγκες ενός προγραμματιστή ή διαχειριστή πληροφοριακού συστήματος.

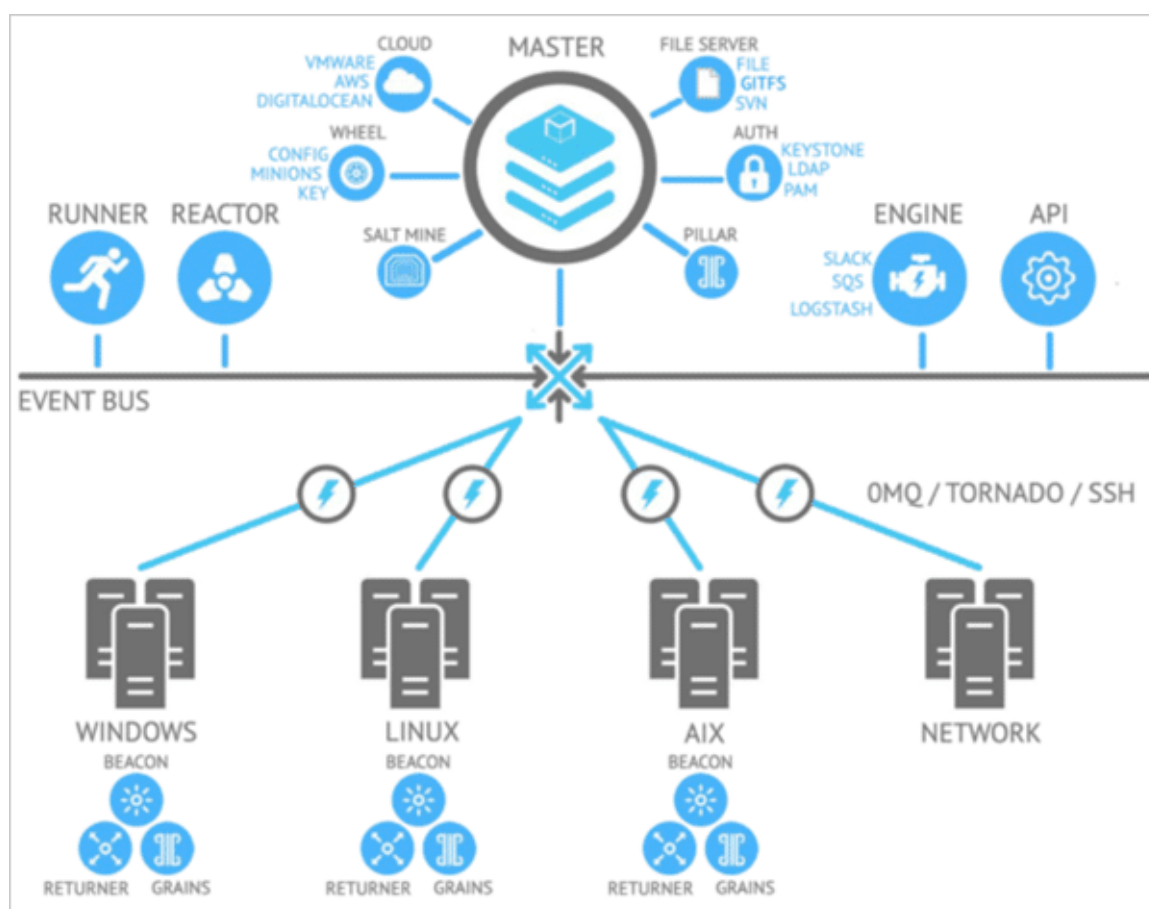
Οι ενότητες εκτέλεσης (execution modules) αντιπροσωπεύουν τις διαθέσιμες λειτουργίες για άμεση εκτέλεση από το μηχανισμό απομακρυσμένης εκτέλεσης. Αυτές οι ενότητες περιέχουν τις συγκεκριμένες πληροφορίες μεταξύ πλατφορμών που χρησιμοποιεί το «Salt» για τη διαχείριση της φορητότητας και αποτελούν τον πυρήνα του περιβάλλοντος προγραμματισμού

³⁶ <https://www.upguard.com/articles/configuration-management-tools>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

εφαρμογών «API» των λειτουργιών επιπέδου συστήματος που χρησιμοποιούνται από τα συστήματα «Salt».

Οι λειτουργικές μονάδες κατάστασης (state modules) είναι τα συστατικά που αποτελούν το «backend» για το σύστημα διαχείρισης διαμόρφωσης «Salt». Αυτές οι ενότητες εκτελούν τον κώδικα που απαιτείται για την επιβολή, τη ρύθμιση ή την αλλαγή της διαμόρφωσης ενός συστήματος - στόχου. Όπως και σε άλλες ενότητες, οι περισσότερες καταστάσεις καθίστανται διαθέσιμες όταν προστίθενται στις ενότητες καταστάσεων³⁷.



Εικόνα 8: Αρχιτεκτονική του εργαλείου «Salt».

³⁷ <https://docs.saltstack.com/en/latest/>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Το σύστημα «Grains» είναι ένα σύστημα για την ανίχνευση στατικών πληροφοριών ενός συστήματος και την αποθήκευσή τους στη μνήμη «RAM» για τη γρήγορη συλλογή τους.

Οι μονάδες «Renderer» χρησιμοποιούνται για την απόδοση των πληροφοριών που διαβιβάζονται στο σύστημα κατάστασης του «Salt». Το σύστημα απόδοσης είναι αυτό που καθιστά δυνατή την αναπαράσταση των δεδομένων διαχείρισης διαμόρφωσης του «Salt», σε οποιαδήποτε σειριοποιήσιμη μορφή.

«Returners»: Οι κλήσεις απομακρυσμένης εκτέλεσης που πραγματοποιούνται από το «Salt» αποσυνδέονται από το σύστημα κλήσεων. Αυτό επιτρέπει την επιστροφή των πληροφοριών που δημιουργούνται από την απομακρυσμένη εκτέλεση σε μια αυθαίρετη τοποθεσία. Η διαχείριση των αυθαίρετων τοποθεσιών επιστροφής διαχειρίζεται από τις μονάδες επιστροφής (Returner Modules).

Οι «runners» είναι εφαρμογές βασικής ευκολίας (master side convenience applications) που εκτελούνται με την εντολή «salt-run»³⁸.

Το «Salt» λειτουργεί σε ένα μοντέλο εγκατάστασης «master – client» ή σε ένα μη κεντρικό μοντέλο. Παρέχει μεθόδους «push» και «SSH – Secure Shell» για την επικοινωνία με τους πελάτες «clients». Τέλος, επιτρέπει την ομαδοποίηση πελατών και προτύπων διαμόρφωσης για τον απλό και εύκολο έλεγχο του περιβάλλοντος. Το «Salt» χρησιμοποιείται από ποικίλους οργανισμούς, όπως οι «JobSpring Partners», «DISH Network Corporation», «Everbridge Inc», «Cloudflare Inc», «Ubisoft S.A.» κλπ.

Στα πλεονεκτήματά του συγκαταλέγεται η απλότητα στη χρήση του, εφόσον εγκατασταθεί. Επιπλέον, διαθέτει δυνατότητα εκμετάλλευσης της γλώσσας «DSL» (Domain Specific Language) και επομένως δεν απαιτεί λογική και καταστάσεις. Επίσης, η είσοδος, η έξοδος και οι ρυθμίσεις του «Salt» χαρακτηρίζονται για τη σταθερότητά και τη συνέπεια τους, αφού χρησιμοποιείται η δηλωτική γλώσσα «YAML».

³⁸ [https://en.wikipedia.org/wiki/Salt_\(software\)](https://en.wikipedia.org/wiki/Salt_(software))

Μειονέκτημα ωστόσο, αποτελεί η αρχική διαδικασία εγκατάστασης, η οποία είναι εξαιρετικά δύσκολη, ειδικά όσον αφορά τις ρυθμίσεις που θα πρέπει να πραγματοποιηθούν αλλά και την κατανόηση του τρόπου εκτέλεσής τους, ιδίως από νέους χρήστες του εργαλείου. Πέραν αυτού, η υποστήριξη για λειτουργικά συστήματα πέραν των «Linux» είναι περιορισμένη³⁹.

³⁹<https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
<https://www.upguard.com/articles/configuration-management-tools>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

2.3.6 Εργαλείο «Ansible»

Το εργαλείο «Ansible» είναι ένα κοινοτικό έργο ανοικτού κώδικα (open source project) που χρηματοδοτείται από την εταιρεία «Red Hat» και αποτελεί τον απλούστερο τρόπο αυτοματοποίησης τεχνολογίας πληροφοριών. Παράλληλα, είναι αναπτυγμένο στη δική του γλώσσα την ομώνυμη «Ansible», η οποία είναι η μόνη γλώσσα αυτοματισμού που μπορεί να χρησιμοποιηθεί σε ολόκληρες ομάδες πληροφορικής «IT», από διαχειριστές δικτύου έως προγραμματιστές και διαχειριστές πληροφοριακών συστημάτων. Βασίζεται στην γλώσσα προγραμματισμού «Python» και στη διεπαφή γραμμής εντολών «PowerShell» των «Windows» και υποστηρίζεται από λειτουργικά συστήματα «Linux», «Unix», «Windows», «MAC OS» κλπ. Ενδεικτικά, κάποιοι μεγάλοι οργανισμοί και επιχειρήσεις που χρησιμοποιούν το «Ansible», είναι οι «Atlassian», «Allegiant», «Cisco», «Gartner», «NASA», «Twitter», «Verizon», «NEC», «Porter» κλπ.

Η λειτουργία του «Ansible» έγκειται στη σύνδεση των επιμέρους κόμβων του συστήματος και στην προώθηση των «Ansible modules» σε αυτούς. Το «Ansible», εκτελεί απομακρυσμένα τα προαναφερόμενα «modules» στον κόμβο που έχει οριστεί, και στη συνέχεια τα διαγράφει από τον εν λόγω κόμβο, εφόσον έχει ολοκληρωθεί η εκτέλεσή τους.

Οι διαδικασίες αυτοματοποίησης που ορίζονται από την «Ansible», ορίζονται μέσα από τα «Ansible Playbooks», χρησιμοποιώντας τη γλώσσα σήμανσης (markup language) «YAML», η σύνταξη της οποίας, παραπέμπει στον ανθρώπινο λόγο.

Στα «Playbooks», αναλύεται η επιθυμητή διαδικασία που πρόκειται να εκτελεστεί, μέσα από ένα σύνολο απλών και κατανοητών βημάτων, τα οποία με τη σειρά τους αποτελούνται από ένα σύνολο μερών εντολών, αποκαλούμενα ως «tasks». Τα «tasks» εκτελούνται με διαδοχική σειρά, ένα κάθε φορά, σε όλα τα μηχανήματα – στόχους που έχουν οριστεί στην αρχή του εκάστοτε «playbook». Είναι σημαντικό να αναφερθεί ότι όλοι οι «hosts» που ορίζονται σε ένα «playbook», θα λάβουν ακριβώς τις ίδιες «οδηγίες» και εντολές προς εκτέλεση⁴⁰.

⁴⁰ https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ένα «playbook» εκτελείται από πάνω προς τα κάτω. Οι υπολογιστές – στόχοι (hosts) στους οποίους απέτυχε η εκτέλεση κάποιου/κάποιων από τα «tasks», εξαιρούνται από την εκτέλεση των υπόλοιπων «tasks» που εμπεριέχονται στο «playbook».

Στόχος είναι σε κάθε «task», να εκτελείται κάποιο «module» με πολύ συγκεκριμένες παραμέτρους (arguments). Σε κάθε «module», μπορούν να χρησιμοποιηθούν μεταβλητές για τον καθορισμό των παραμέτρων αυτών.

Τα «modules» που επιλέγονται προς εκτέλεση, θα πρέπει να είναι σχετικά ομοιογενή, έτσι ώστε η εκτέλεση ενός «module» πολλές φορές σε μια σειρά, να έχει το ίδιο αποτέλεσμα με την εκτέλεση του μια μόνο φορά. Ένας τρόπος για να επιτευχθεί αυτό, είναι ο έλεγχος εάν με την εκτέλεση ενός «module» έχει ήδη επιτευχθεί η επιθυμητή τελική κατάσταση και σε θετική περίπτωση, να μην εκτελεστεί κάποια επιπλέον ενέργεια. Εφόσον, όλες οι λειτουργικές μονάδες που χρησιμοποιεί ένα «playbook» είναι ομοιογενείς, τότε μπορεί να πιθανολογηθεί ότι το ίδιο το «playbook» είναι ομοιογενές. Επομένως, η εκ νέου εκτέλεση του «playbook» θα μπορεί κατ' επέκταση να χαρακτηριστεί ασφαλής.

Σε ό,τι αφορά τα «Windows modules» του «Ansible», αυτά ουσιαστικά αποτελούν έτοιμα «κομμάτια κώδικα», αναπτυγμένα από τα «cmd-lets» της διεπαφής «PowerShell» των λειτουργικών συστημάτων «Windows». Έτσι, «καλώντας» σε ένα «playbook» κάποιο από τα «modules» αυτά, ουσιαστικά εκτελείται μια σειρά εντολών και διαδικασιών με σκοπό την επίτευξη του επιθυμητού αποτελέσματος, το οποίο εξαρτάται από το εάν αυτά συνδυαστούν σωστά και με την κατάλληλη διαδοχική σειρά⁴¹.

Το «Ansible» συνδέεται γενικά μέσω «SSH» με όλα τα υπόλοιπα λειτουργικά (linux, unix, κα), που υποστηρίζουν το εργαλείο, πλην αυτών των «Windows», όπου μέσω της υπηρεσίας «WinRM» (Windows Remote Management) εκτελεί τις εντολές ή τα προκαθορισμένα «modules» του εργαλείου για τα «Windows». Τα εν λόγω «modules» αποτελούνται από ένα σύνολο «cmd-lets», τα οποία εκτελούνται απομακρυσμένα μέσω της διεπαφής «Powershell»

⁴¹ https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html

των «Windows». Με τους τρόπους αυτούς, εκλείπει η ανάγκη χρήσης κάποιου «agent» για την εδραίωση της επικοινωνίας με το εκάστοτε υπολογιστικό σύστημα⁴².

Σε αντίθεση με τα περισσότερα λογισμικά διαχείρισης διαμόρφωσης, το «Ansible» δεν απαιτεί παρά μόνο ένα μηχανήμα ελέγχου από όπου ξεκινά η «ενορχήστρωση» (orchestration). Το «Ansible» λειτουργεί έναντι πολλών συστημάτων στην υποδομή πληροφορικής, επιλέγοντας τμήματα του αρχείου «inventory» του «Ansible», αποθηκευμένα ως αρχεία κειμένου «ASCII» με δυνατότητα επεξεργασίας και έκδοσης. Επιπλέον, όχι μόνο το προαναφερόμενο αρχείο «inventory» είναι διαμορφώσιμο, αλλά είναι δυνατόν, επίσης, να χρησιμοποιηθούν πολλά αρχεία του «inventory» ταυτόχρονα, τα οποία μπορεί να αντλούνται από δυναμικές πηγές ή πηγές υπολογιστικού νέφους (cloud) ή σε διαφορετικές μορφές («YAML», «INI», κ.λπ.)⁴³.

Οποιοδήποτε μηχανήμα, με εγκατεστημένα βοηθητικά προγράμματα «Ansible», μπορεί να χρησιμοποιήσει ένα σύνολο αρχείων / καταλόγων με σκοπό να ενορχηστρώσει άλλους κόμβους. Η απουσία απαίτησης κεντρικού εξυπηρετητή, απλοποιεί σημαντικά τον σχεδιασμό αποκατάστασης καταστροφών (disaster recovery). Η διαχείριση των κόμβων γίνεται από έναν υπολογιστή ελέγχου - συνήθως μέσω «SSH» ή/και της υπηρεσίας «WinRM». Η μηχανή ελέγχου (control machine) περιγράφει τη θέση των κόμβων μέσω του αρχείου «inventory» της. Τα «ευαίσθητα» δεδομένα μπορούν να αποθηκευτούν σε κρυπτογραφημένα αρχεία, χρησιμοποιώντας το «Ansible Vault», το οποίο βρίσκεται σε κυκλοφορία από το 2014.

Σε αντίθεση με άλλα δημοφιλή λογισμικά διαχείρισης διαμόρφωσης, όπως «Chef», «Puppet» και «CFEngine», το «Ansible» χρησιμοποιεί αρχιτεκτονική χωρίς κάποιον «agent», δηλαδή χωρίς να εγκαθίσταται κάποιο συγκεκριμένο λογισμικό για τη χρήση του «Ansible» στον ελεγχόμενο κόμβο. Αντ' αυτού, το «Ansible» ενορχηστρώνει έναν κόμβο εγκαθιστώντας και εκτελώντας μονάδες στον κόμβο προσωρινά μέσω «SSH» ή/και της υπηρεσίας «WinRM». Κατά τη διάρκεια μιας εργασίας, μια διαδικασία που εκτελεί τη μονάδα (module), επικοινωνεί με τη μηχανή ελέγχου με ένα πρωτόκολλο που βασίζεται σε μορφότυπο «JSON» μέσω της τυπικής εισόδου και εξόδου του. Όταν το «Ansible» δεν διαχειρίζεται έναν κόμβο, δεν καταναλώνει πόρους στον

⁴² [https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))· <https://www.ansible.com/overview/it-automation>,

⁴³ <https://www.upguard.com/articles/configuration-management-tools>

κόμβο, επειδή δεν εκτελείται καμία υπηρεσία («service»/»daemon») εγκατεστημένου λογισμικού.

Το «Ansible» αποτελεί ένα από τα πιο αξιόλογα εργαλεία διαχείρισης διαμόρφωσης, ανάπτυξης, ενορχηστρώσεων, καθώς και αυτοματισμού. Αποτελεί εργαλείο βασισμένο στη μέθοδο «push» (push – based tool) και βοηθά στην αυτοματοποίηση ολόκληρης της υποδομής πληροφορικής ενός οργανισμού ή μιας επιχείρησης, συμβάλλοντας στη αύξηση της παραγωγικότητας και προσδίδοντας μεγαλύτερα κέρδη στον οργανισμό ή την επιχείρηση.

Στα πλεονεκτήματά του, όπως αναφέρθηκε και παραπάνω, συγκαταλέγεται το γεγονός ότι είναι «agentless» και δεν απαιτείται κάποια εγκατάσταση στον ελεγχόμενο κόμβο, διότι βασίζεται σε συνδέσεις μέσω «SSH» ή/και της υπηρεσίας «WinRM», καθώς επίσης ότι έχει δημιουργηθεί μία ισχυρή κοινότητα, έτοιμη για σχετική υποστήριξη και επίλυση τυχόν ζητημάτων. Επιπρόσθετα, η δομή των «playbooks» είναι ιδιαίτερα ισχυρή και η σύνταξη τους σχετικά εύκολη, αφού δεν απαιτούνται εξειδικευμένες γνώσεις προγραμματισμού. Το «Ansible» έχει, επίσης, μια πιο απλοποιημένη βάση κώδικα σε σύγκριση με κάποια άλλα εργαλεία και προσφέρει υψηλή διαλειτουργικότητα. Τέλος, πληροί τις προϋποθέσεις ασφαλείας όσον αφορά τις συνδέσεις τις οποίες παρέχει, αφού όπως προαναφέρθηκε αυτές πραγματοποιούνται μέσω «SSH» σε λοιπά περιβάλλοντα και μέσω της υπηρεσίας «WinRM» σε περιβάλλοντα «Windows», χρησιμοποιώντας σε αμφότερα το πρωτόκολλο «HTTPS» (Hypertext Transfer Protocol Secure)⁴⁴.

Μειονέκτημα αποτελεί, ωστόσο, για το «Ansible» ότι είναι λιγότερο αποτελεσματικό από άλλα εργαλεία, τα οποία βασίζονται σε άλλες γλώσσες προγραμματισμού. Επιπλέον, πραγματοποιεί τη λογική τροποποίησή του μέσω της γλώσσας «DSL», γεγονός που απαιτεί τον συνεχή έλεγχο της σχετικής τεκμηρίωσης (documentation) μέχρι την εκμάθηση του εργαλείου. Στο «Ansible», επίσης, ζητείται εγγραφή μεταβλητής ακόμη και στις πιο απλές λειτουργίες, με αποτέλεσμα

⁴⁴<https://www.ansible.com/overview/it-automation>·
<https://www.ansible.com/overview/how-ansible-works?intcmp=701f2000000h4RcAAI>·
[https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))·
<https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

εύκολες εργασίες να μετατρέπονται σε περισσότερο περίπλοκες. Τέλος, η δοκιμή ανάπτυξης των «modules» του εργαλείου «Ansible», θα μπορούσε να χαρακτηριστεί σχετικά φτωχή.

Στο σημείο αυτό, αξίζει να αναφερθεί η ύπαρξη του «Ansible Tower», το οποίο δεν παρέχεται δωρεάν, υποστηρίζει και αυτό τα ίδια λειτουργικά συστήματα, παρέχει διεπαφή ιστού «web interface», και μπορεί να εγκατασταθεί σε κάποιον εξυπηρετητή ή σε υποδομή υπολογιστικού νέφους (cloud infrastructure), όπως για παράδειγμα «AWS»). Παρέχει τη δυνατότητα δημιουργίας των «playbooks» μέσω της διεπαφής (interface), τη δημιουργία προτύπων (templates) εργασιών, στατιστικά δεδομένα κλπ. Το «Red Hat Ansible Tower» επιτρέπει τη συγκέντρωση και τον έλεγχο της υποδομής πληροφορικής μέσω ενός οπτικού πίνακα ελέγχου (dashboard), τον έλεγχο της πρόσβασης βάσει ρόλων (role based) και πολλά άλλα⁴⁵.

2.4 Σύγκριση Δυνατοτήτων - Επιλογή

Όλα τα ανωτέρω περιγραφόμενα εργαλεία, αποτελούν εργαλεία «διαχείρισης διαμόρφωσης», που σημαίνει ότι έχουν σχεδιαστεί για την ανάπτυξη, διαμόρφωση και διαχείριση εξυπηρετητών και γενικότερα υποδομών πληροφορικής. Υπάρχουν πολλοί παράγοντες, οι οποίοι μπορούν να ληφθούν υπόψη και να εξεταστούν για τη σύγκριση τους και την επιλογή του κατάλληλου εργαλείου για την εκπόνηση των μελετών περίπτωσης της παρούσας εργασίας.

Στην εν λόγω διπλωματική εργασία, εξετάζεται, η αυτοματοποιημένη παραμετροποίηση σε περιβάλλον «Windows», οπότε ένα από τα βασικά χαρακτηριστικά που θα πρέπει να ληφθεί υπόψη είναι η διαλειτουργικότητα. Άλλες πολύ σημαντικές παράμετροι που θα πρέπει να συνυπολογιστούν, είναι η αρχική εγκατάσταση και παραμετροποίηση των εργαλείων αυτών, η μετέπειτα χρήση και διαχείριση του εκάστοτε εργαλείου, η επεκτασιμότητα, η διαθεσιμότητα, η ασφάλεια (εξετάζεται εάν το κάθε εργαλείο από τα παραπάνω παρέχει κάποια μέτρα ασφάλειας ή κάποιον τρόπο κρυπτογράφησης), το αν απαιτείται κάποιος «agent» στους ελεγχόμενους κόμβους, η γλώσσα που χρησιμοποιεί το κάθε εργαλείο, καθώς επίσης και εάν

⁴⁵ <https://www.ansible.com/products/tower>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

υφίσταται υποστήριξη σε περιβάλλοντα «Windows». Για το σκοπό αυτό και με βάση όσα αναπτύχθηκαν αναλυτικά στην [παράγραφο 2.1](#) της παρούσας μελέτης, καταρτίσθηκε ο παρακάτω πίνακας προκειμένου να γίνουν ευκολότερα αντιληπτά τα δεδομένα για την επιλογή.

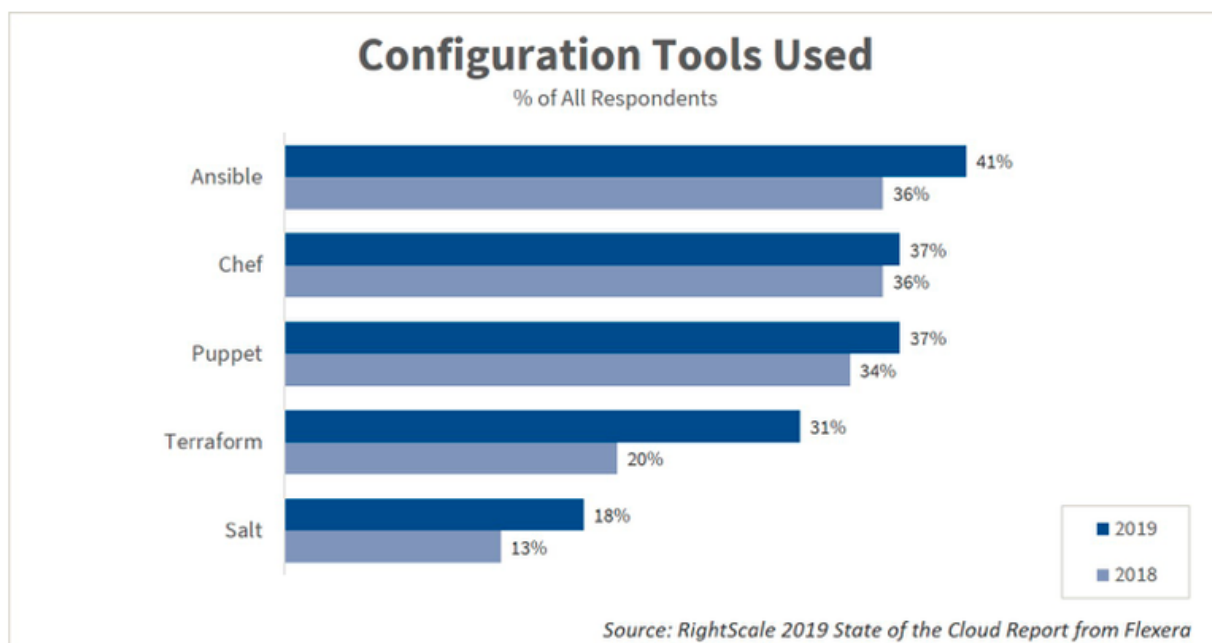
Πίνακας 3: Συγκριτικός πίνακας εργαλείων αυτοματοποιημένης διαχείρισης διαμόρφωσης.

Μέγεθος	CFEngine	Rudder	Puppet	Chef	Saltstack	Ansible
Διαθεσιμότητα	Υψηλή Διαθεσιμότητα	Υψηλή Διαθεσιμότητα	Υψηλή Διαθεσιμότητα	Υψηλή Διαθεσιμότητα	Υψηλή Διαθεσιμότητα	Υψηλή Διαθεσιμότητα
Αρχική Εγκατάσταση	Αρκετά Πολύπλοκη	Πολύπλοκη	Πολύπλοκη	Πολύπλοκη	Πολύπλοκη	Εύκολη
Διαχείριση	Πολύπλοκη	Πολύπλοκη	Πολύπλοκη	Πολύπλοκη	Εύκολη	Εύκολη
Επεκτασιμότητα	Υψηλή Επεκτασιμότητα	Υψηλή Επεκτασιμότητα	Υψηλή Επεκτασιμότητα	Υψηλή Επεκτασιμότητα	Υψηλή Επεκτασιμότητα	Υψηλή Επεκτασιμότητα
Γλώσσα Διαμόρφωσης	DSL(CFEngine language), C	C, Scala	DSL(Puppet DSL Ruby)	DSL(Ruby)	YAML(Python)	YAML(Python)/Power Shell
Διαλειτουργικότητα	Υψηλή	Υψηλή	Υψηλή	Υψηλή	Υψηλή	Υψηλή
Κρυπτογράφηση	Ναι	Ναι	Ναι	Ναι	Ναι	Ναι
Agentless	Όχι	Όχι	Όχι	Όχι	Όχι	Ναι
Υποστήριξη Windows	Ναι	Ναι	Ναι	Ναι	Ναι	Ναι (απαιτείται μηχανήμα ελέγχου linux)

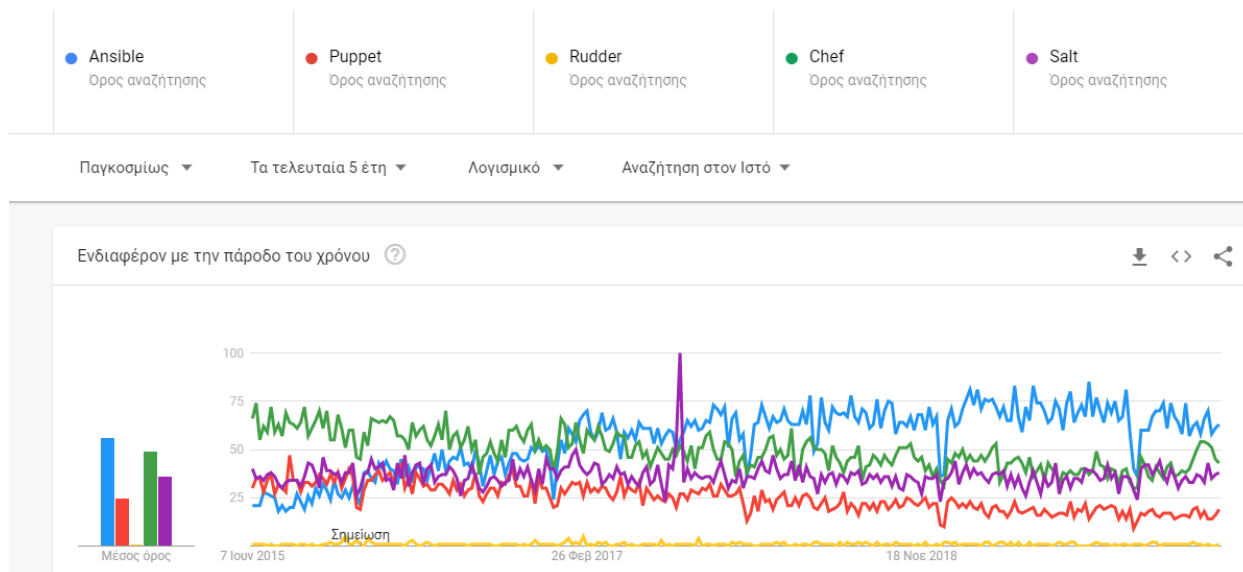
Πέραν των μεριδίων αγοράς που περιγράφηκαν προηγουμένως, αξίζει, επιπλέον, να γίνει μνεία στη χρήση και τις τάσεις χρήσης των ανωτέρω εργαλείων.

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή



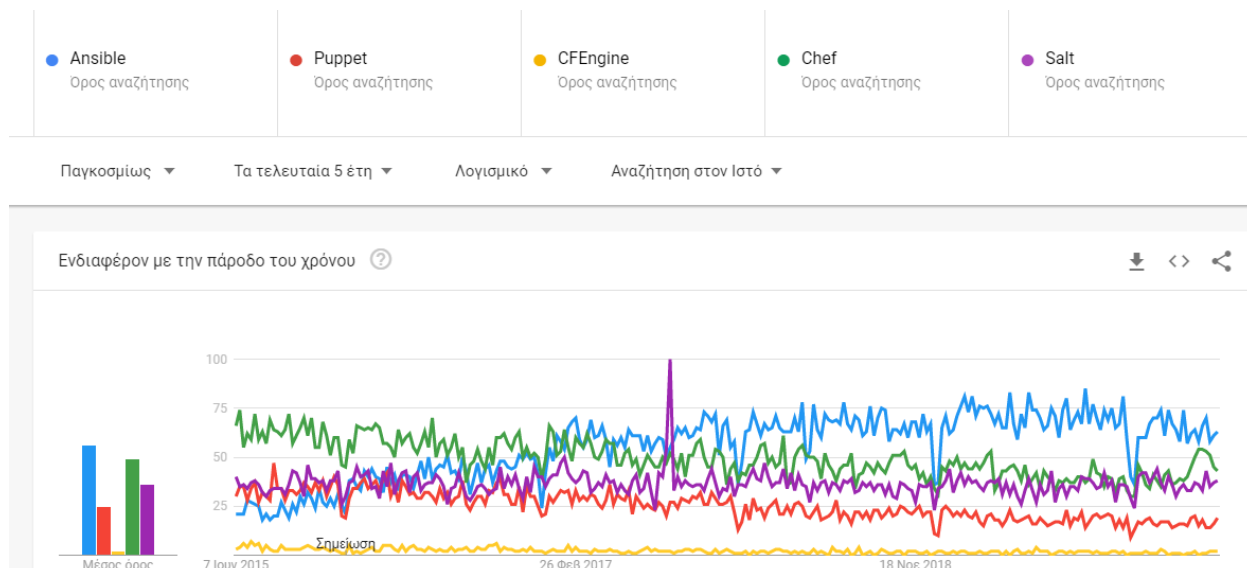
Εικόνα 9: Χρήση των δημοφιλέστερων εργαλείων αυτοματοποιημένης διαχείρισης διαμόρφωσης ανά έτος. (Πηγή: <https://www.techrepublic.com/article/ansible-overtakes-chef-and-puppet-as-the-top-cloud-configuration-management-tool/>)



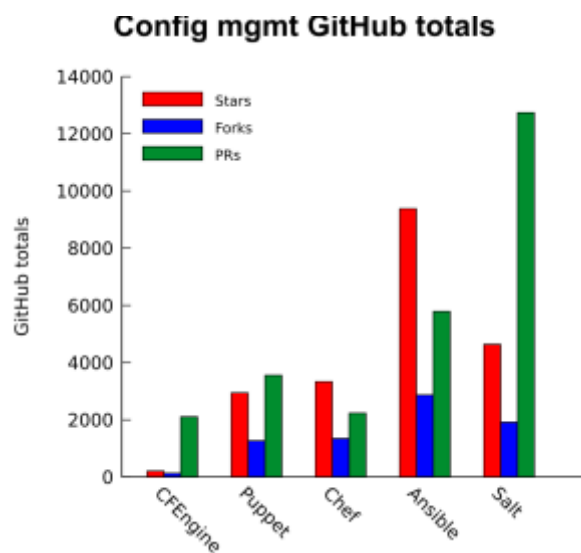
Εικόνα 10: Τάση χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης τα τελευταία πέντε (5) έτη. (Πηγή: Google trends)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή



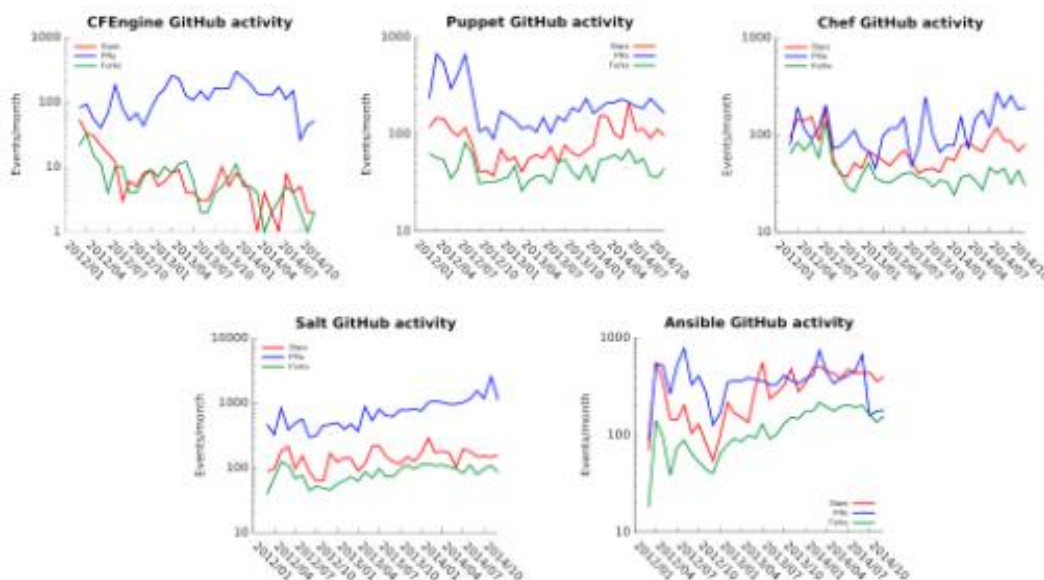
Εικόνα 11: Τάση χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης τα τελευταία πέντε (5) έτη. (Πηγή: Google trends)



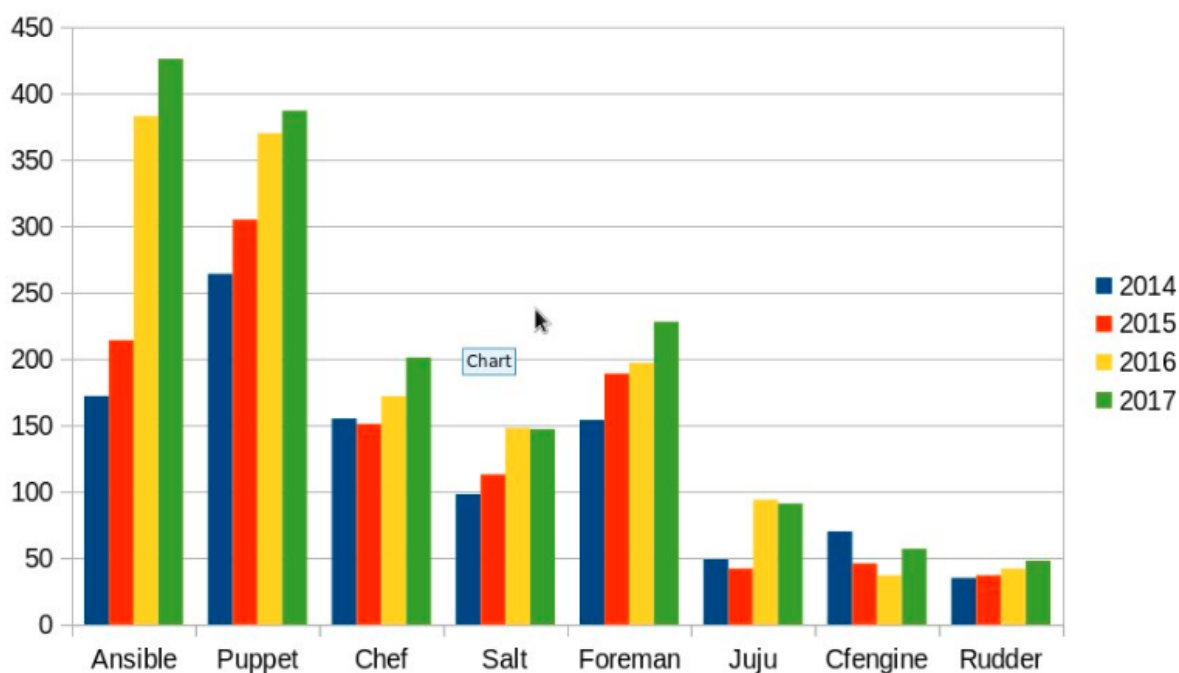
Εικόνα 12: Σύνολα «pull requests» των δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης στον ιστότοπο «GitHub». (Πηγή: <https://redmonk.com/dberkholz/2015/04/02/quantifying-configuration-management-communities/>)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή



Εικόνα 13: Σύγκριση και τάσεις χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης στον ιστότοπο «GitHub». (Πηγή: <https://redmonk.com/dberkholz/2015/04/02/quantifying-configuration-management-communities/>)



Εικόνα 14: Τάσεις χρήσης δημοφιλέστερων εργαλείων διαχείρισης διαμόρφωσης ανά έτος. (Πηγή: <https://lvops.com/news/ansible/64.html>)

Για την εκπόνηση της παρούσας μελέτης, επιλέχθηκε το εργαλείο «Ansible», διότι αποτελεί ένα εργαλείο εύχρηστο, το οποίο δεν απαιτεί τη διάθεση υψηλών πόρων, μπορεί να λειτουργήσει σε εικονικά περιβάλλοντα, σε περιβάλλοντα υπολογιστικού νέφους (cloud) και σε περιβάλλοντα

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

υβριδικού υπολογιστικού νέφους (hybrid cloud). Επιπρόσθετα, υποστηρίζει ικανοποιητικά και την παραμετροποίηση λειτουργικών συστημάτων «Windows», τα οποία αποτελούν και το αντικείμενο μελέτης της παρούσας διπλωματικής εργασίας.

Το εργαλείο «Ansible» έχει ως βασικό πλεονέκτημα, το γεγονός ότι δεν απαιτεί εγκατάσταση «agent» στους υπολογιστές - στόχους (hosts), στα οποία θα γίνει οποιαδήποτε αυτοματοποιημένη παραμετροποίηση, όπως εγκατάσταση λογισμικού, διαχείριση υπηρεσιών (services) και διαφόρων πολιτικών ασφαλείας, εισαγωγή σε δίκτυο τομέα (domain network), καθώς και οτιδήποτε άλλο απαιτείται να γίνει σχετική ανάπτυξη (deploy). Με την «agentless» αρχιτεκτονική του, η χρήση του «Ansible» σε ένα καινούριο σύστημα προσφέρει αμεσότητα, αλλά και εξοικονόμηση πόρων, τόσο του δικτύου όσο και του συστήματος εν γένει.

Η υπηρεσία «WinRM» μέσω της οποίας καθίσταται δυνατή η επικοινωνία μεταξύ του μηχανήματος ελέγχου του «Ansible» (Ansible Control machine) και των υπολογιστών με λειτουργικά συστήματα «Windows», στους οποίους πραγματοποιείται και η μελέτη, αποτελεί υπηρεσία (service) των συστημάτων «Windows», με σκοπό την απομακρυσμένη επικοινωνία και την αλληλεπίδραση με άλλα λογισμικά, εφαρμογές κλπ. Το «Ansible» αλληλεπιδρά με τα «Windows» μέσω «modules» που χρησιμοποιούν ουσιαστικά, εντολές για περιβάλλοντα «Windows», οι οποίες εκτελούνται μέσω της διεπαφής «PowerShell» των «Windows».

Η υπηρεσία «WinRM» θεωρεί το κανάλι επικοινωνίας κρυπτογραφημένο εάν χρησιμοποιεί πιστοποιητικό «TLS» με τη χρήση πρωτοκόλλου «HTTP» (HTTPS) ή χρησιμοποιώντας κρυπτογράφηση σε επίπεδο μηνύματος. Η χρήση της υπηρεσίας «WinRM» με «TLS» αποτελεί επιλογή που συνιστάται, καθώς λειτουργεί με όλες τις επιλογές ελέγχου ταυτότητας. Παρ' όλ' αυτά, απαιτεί τη δημιουργία και χρήση πιστοποιητικού στον «ακροατή WinRM» (WinRM listener)⁴⁶.

Στον «ακροατή WinRM» (WinRM listener) στον υπολογιστή – πελάτη (host) ορίστηκε το «HTTPS transport» με την προεπιλεγμένη από τα «Windows» θύρα 5986 (default port), η οποία αποτελεί

⁴⁶ https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html.

και την προεπιλεγμένη θύρα, που χρησιμοποιεί ο ακροατής του πρωτοκόλλου «HTTPS» στα λειτουργικά συστήματα «Windows».

Σε ό,τι αφορά την ασφάλεια, μπορεί επιπλέον να γίνει χρήση του «Ansible Vault», το οποίο αποτελεί έναν χώρο, ο οποίος λειτουργεί σαν «θησαυροφυλάκιο». Ουσιαστικά είναι ένα αρχείο, στο οποίο φυλάσσονται «ευαίσθητα» δεδομένα, όπως κωδικοί πρόσβασης, «διευθύνσεις IP» από κρίσιμα μηχανήματα με τα οποία αλληλεπιδρά το «Ansible», ονόματα χρηστών κλπ., και δεν είναι εφικτό αυτά τα στοιχεία να προσπελαστούν από κάποιον άλλον χρήστη, πλην του διαχειριστή του μηχανήματος ελέγχου (control machine) του «Ansible». Η χρήση του «Ansible Vault» θα αναλυθεί περαιτέρω παρακάτω.

Ένα δεύτερο πλεονέκτημα του εργαλείου «Ansible», είναι ότι η δηλωτική γλώσσα που χρησιμοποιείται για την σύνταξη των «playbooks», η «YAML», διαθέτει καθαρή δομή και ομαλή καμπύλη εκμάθησης.

Κεφάλαιο 3^ο: Ορισμός του Προβλήματος και Τεχνική Ανάπτυξη

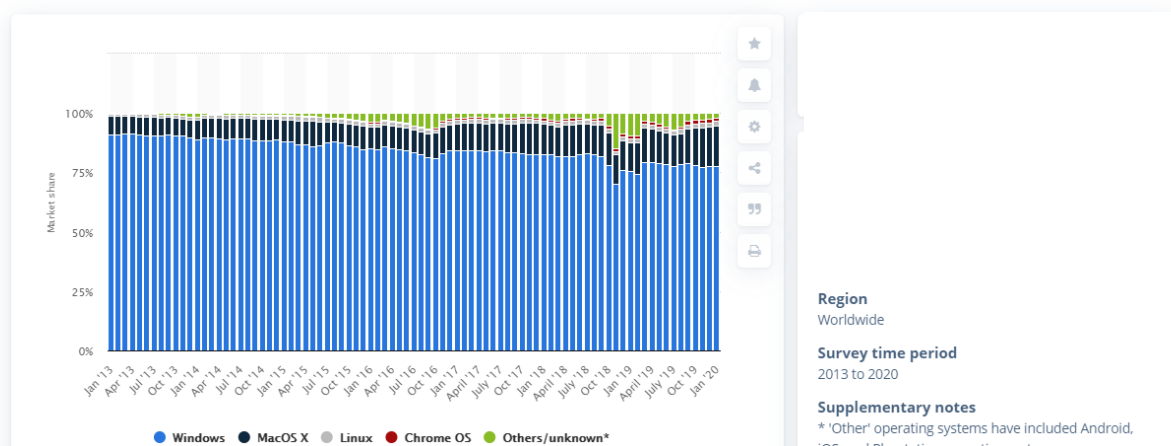
3.1 Ορισμός του Προβλήματος

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η αυτοματοποιημένη παραμετροποίηση συστημάτων Ηλεκτρονικών Υπολογιστών, οι οποίοι λειτουργούν με λειτουργικά συστήματα διαφόρων εκδόσεων των «Windows», χρησιμοποιώντας το εργαλείο «Ansible» και τις δυνατότητές του.

Είναι γεγονός, ότι τα λειτουργικά συστήματα «Microsoft Windows» εισήλθαν και συνέχισαν να διατηρούν κυρίαρχη παρουσία, στην αγορά του λειτουργικού συστήματος επιτραπέζιων υπολογιστών, από το 1985. Παρά την ελαφρά απώλεια μεριδίου αγοράς τα τελευταία χρόνια, το λειτουργικό σύστημα των «Windows» παραμένει πρωτοπόρος με υψηλό μερίδιο 77,7%.

Technology & Telecommunications > Software

Global market share held by operating systems for desktop PCs, from January 2013 to January 2020



Εικόνα 15: Παγκόσμιο μερίδιο της αγοράς λειτουργικών συστημάτων επιτραπέζιων υπολογιστών⁴⁷.

Συνεπώς, κρίνεται απαραίτητη η μελέτη μεθόδων αυτοματοποιημένων παραμετροποιήσεων που αφορούν στα λειτουργικά συστήματα «Windows», καθόσον όπως έχει ήδη αναφερθεί η πλειονότητα των υποδομών των πληροφορικών συστημάτων των σημερινών οργανισμών και επιχειρήσεων περιλαμβάνουν σε μεγάλο βαθμό λειτουργικά συστήματα «Windows».

⁴⁷<https://www.statista.com/statistics/218089/global-market-share-of-windows-7/#statisticContainer>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Στη συνέχεια, θα μελετηθούν διαδικασίες για την υλοποίηση των οποίων αφενός απαιτείται η αφιέρωση αρκετού χρόνου από τους διαχειριστές υποδομών πληροφορικής, λόγω της αναγκαιότητας διαρκούς επανάληψής τους, αφετέρου δε πιθανώς να απαιτούν τη φυσική τους παρουσία. Τέτοιες εργασίες συνιστούν, λόγου χάριν, η εγκατάσταση κάποιου λογισμικού, καθώς και η εισαγωγή ή εξαγωγή ενός ηλεκτρονικού υπολογιστή σε ένα δίκτυο τομέα (domain network).

Τέτοιου είδους διαδικασίες είναι δυνατό να υλοποιηθούν μέσω των δυνατοτήτων που παρέχει ο «Windows Server», ο οποίος διαθέτει τον ρόλο του «Domain Controller» σε μια υποδομή, με τη χρήση κάποιων Αντικειμένων Πολιτικών Ομάδας (Group Policy Objects) ή με τη χρήση κάποιων αρχείων δεσμών εντολών (scripts), που εκτελούνται στους υπολογιστές - πελάτες του δικτύου (Windows hosts), με το εργαλείο του «Group Policy Manager». Με τις διαδικασίες αυτές δεν είναι πάντοτε εφικτή η πλήρης παραμετροποίηση μιας υποδομής. Το πρόβλημα αυτό μπορεί να αντιμετωπιστεί με τη χρήση του εργαλείου «Ansible», όπου μια διαδικασία όπως είναι η εγκατάσταση ενός πακέτου λογισμικού μπορεί όχι μόνο να παραμετροποιηθεί πλήρως, αλλά και να πραγματοποιηθεί αυτοματοποιημένα και απομακρυσμένα είτε στο σύνολο ή μέρος των εξυπηρετητών της υποδομής, είτε στο σύνολο ή μέρος των υπολογιστών – πελατών αυτής.

Τα «modules» που παρέχει το «Ansible» για την αυτοματοποιημένη παραμετροποίηση και τη διαχείριση λειτουργικών συστημάτων «Windows», μπορούν να συνδυαστούν σε ένα αρχείο «playbook», το οποίο θα εκτελεστεί σε ένα περιβάλλον δικτύου τομέα (domain network), ή σε ένα κομμάτι αυτού, ή ακόμη και σε μεμονωμένους υπολογιστές – στόχους (hosts), με σκοπό να επιτευχθεί το επιθυμητό αποτέλεσμα. Με τον τρόπο αυτό, ελαττώνεται σε πολύ μεγάλο βαθμό η ανθρώπινη παρέμβαση για την πραγματοποίηση των προαναφερόμενων διαδικασιών και ελαχιστοποιείται ο χρόνος υλοποίησης επαναλαμβανόμενων εργασιών, με αποτέλεσμα τη μείωση σφαλμάτων και την ελαχιστοποίηση περιθωρίων λάθους, ειδικά σε αρκετά μεγάλες υποδομές⁴⁸.

Άλλες διαδικασίες που μπορεί να αποτελέσουν τροχοπέδη στην παραγωγικότητα ενός τμήματος Πληροφορικής, είναι η ένταξη/απομάκρυνση (join/unjoin) ενός ή/και περισσοτέρων

⁴⁸ <https://www.ansible.com/>, <https://www.ansible.com/overview/how-ansible-works>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

υπολογιστών (hosts) σε ένα δίκτυο τομέας (domain network). Η διαδικασία αυτή, απαιτεί την ανθρώπινη παρέμβαση, όμως με τη χρήση του «Ansible» μπορεί να πραγματοποιηθεί αυτόματα με απομακρυσμένο τρόπο.

Η συγκεκριμένη εργασία πραγματοποιείται τον τρόπο με τον οποίο διαδικασίες, οι οποίες απαιτούν ανθρώπινη παρέμβαση από τον διαχειριστή υποδομής πληροφορικής, μέσω της χρήσης του εργαλείου «Ansible», θα υλοποιηθούν αυτοματοποιημένα. Εξετάζονται η εγκατάσταση πακέτων λογισμικού, καθώς επίσης η ένταξη και η απομάκρυνση ενός υπολογιστή με λειτουργικό σύστημα «Windows» (windows host) σε ένα δίκτυο τομέα «Windows» (Windows domain network). Στη συνέχεια αναλύονται οι διαδικασίες που εφαρμόστηκαν, καθώς επίσης, η δημιουργία και η εφαρμογή ενός περιβάλλοντος δοκιμών των εργασιών που διεξήχθησαν.

3.2 Αρχιτεκτονική Περιβάλλοντος Δοκιμών - Παραδοχές

Για την εκπόνηση της συγκεκριμένης διπλωματικής εργασίας, διαμορφώθηκε ένα τοπικό «εργαστήριο», αποτελούμενο από έναν φορητό ηλεκτρονικό υπολογιστή στον οποίο εγκαταστάθηκε το λογισμικό «Workstation Pro», το οποίο παρέχεται από την εταιρεία «VMware», στην πιο πρόσφατη έκδοσή του. Το «VMware Workstation» αποτελεί έναν «Hypervisor» που εκτελείται σε «x64» εκδόσεις των λειτουργικών συστημάτων «Windows» και «Linux» και επιτρέπει στους χρήστες του να εγκαταστήσουν εικονικές μηχανές σε ένα μόνο φυσικό μηχάνημα και να τις χρησιμοποιήσουν παράλληλα με τις υπόλοιπες εργασίες του φυσικού μηχανήματος⁴⁹.

Με τη χρήση του λογισμικού «VMware Workstation», δημιουργήθηκαν τρεις (3) εικονικές μηχανές (Virtual Machines). Μία εικονική μηχανή «Virtual Machine - Vm» με λειτουργικό σύστημα «Windows Server 2016», μία εικονική μηχανή «Vm» με λειτουργικό σύστημα «Windows 10 x64 bit» έκδοση και μία εικονική μηχανή «Virtual Machine - Vm» με λειτουργικό σύστημα «Debian 10 Buster».

⁴⁹ <https://www.vmware.com/products/workstation-pro.html>

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, έγιναν κάποιες παραδοχές οι οποίες κρίνεται αναγκαίο να αναφερθούν και έχουν ως εξής:

1. Προκειμένου να επιτευχθεί η απρόσκοπτη και εύρυθμη, ωστόσο, λειτουργία του εν λόγω «εργαστηρίου», κρίνεται αναγκαίο να ληφθούν υπόψη οι φυσικοί πόροι που παρέχονται από τον φυσικό υπολογιστή, ο οποίος μέσω του «VMWare Workstation», φιλοξενεί τα εικονικά μηχανήματα. Συνεπώς, δεδομένου ότι ο φυσικός υπολογιστής διαθέτει 8 GB μνήμη «RAM» και 250 GB χώρο αποθήκευσης στον σκληρό δίσκο και επεξεργαστή τεσσάρων πυρήνων με χαρακτηριστικά «Intel(R)_Core(TM)_i5-5300U_CPU_@_2.30GHz», δεν θα πρέπει όλοι αυτοί οι πόροι να δοθούν προς χρήση στις εικονικές μηχανές, έτσι ώστε να εξασφαλισθεί η ορθή λειτουργία του φυσικού υπολογιστή που τις φιλοξενεί.
2. Υπάρχει μία εικονική μηχανή με λειτουργικό σύστημα «Linux» διανομής «Debian», έκδοσης 10 (Debian 10 Buster) στο οποίο είναι εγκατεστημένο το «Ansible» και κατέχει τον ρόλο του μηχανήματος ελέγχου «Ansible» (Ansible control machine). Στη συγκεκριμένη μελέτη, η εικονική μηχανή αυτή αποτελεί ένα «Debian 10 Linux VM».
3. Έχει ρυθμιστεί ένα αρχείο «inventory» (Ansible inventory) βάσει των εικονικών μηχανών με λειτουργικό σύστημα «Windows». Στη συγκεκριμένη εργασία, στο αρχείο «inventory» αυτό έχουν καταχωρηθεί μεταβλητές που σχετίζονται με την επικοινωνία μεταξύ του μηχανήματος ελέγχου «Ansible» (Ansible control machine) και των εικονικών μηχανών που περιέχουν λειτουργικό «Windows», οι οποίες περιγράφονται στη συνέχεια.
4. Υπάρχουν προς παραμετροποίηση δύο (2) εικονικές μηχανές με λειτουργικά συστήματα «Windows Server 2016» και «Windows 10» αντίστοιχα. Αυτό κρίνεται σκόπιμο να αναφερθεί, διότι κάποια από τα «cmdlets» που χρησιμοποιήθηκαν στις παραπάνω εικονικές μηχανές, προκειμένου να επιτευχθεί η επικοινωνία μεταξύ αυτών και της εικονικής μηχανής που περιέχει το «Ansible», δεν λειτουργούν με παλαιότερες εκδόσεις των λειτουργικών «Windows».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

5. Αρχικά, η εικονική μηχανή με λειτουργικό σύστημα «Windows 10», δεν βρίσκεται σε κάποιο δίκτυο τομέα (domain network). Τα παραδείγματα και οι μελέτες περίπτωσης εγκατάστασης λογισμικού, που περιγράφονται στη συνέχεια της παρούσας διπλωματικής εργασίας, πραγματοποιήθηκαν ουσιαστικά, σε έναν υπολογιστή ο οποίος δεν αποτελεί μέλος κάποιου δικτύου τομέα (non-domain-joined host). Ωστόσο, οι όποιες παραμετροποιήσεις και διαδικασίες αυτοματοποίησης, μπορούν να πραγματοποιηθούν και να λειτουργήσουν και σε ένα οποιοδήποτε περιβάλλον δικτύου τομέα (domain network), συμπεριλαμβανομένων και των υπολογιστών – πελατών του δικτύου αυτού, παραμετροποιώντας κατάλληλα τις μεταβλητές του αρχείου «Ansible inventory» ή/και εντάσσοντας (join) το μηχάνημα ελέγχου του «Ansible» (Ansible control machine) στο δίκτυο τομέα «Windows» (Windows domain).
6. Για τις μελέτες περίπτωσης εγκατάστασης λογισμικού με τη χρήση του «Ansible», χρησιμοποιείται ο τοπικός διαχειριστής (local administrator) της εικονικής μηχανής με λειτουργικό σύστημα «Windows 10». Η χρήση του λογαριασμού αυτού μπορεί να γίνει είτε καταχωρώντας τον στο αρχείο «Ansible inventory», είτε με την προσθήκη του «ad-hoc» όταν πραγματοποιείται η εκτέλεση ενός «Ansible playbook», χρησιμοποιώντας την παράμετρο, «-u **‘εδώ μπαίνει το username του local administrator’** –ask-pass **windows**». Ένα παράδειγμα: «**ansible -m win_command -a “ipconfig” -u administrator –ask-pass windows**». Στο παράδειγμα αυτό, εκτελείται από το μηχάνημα ελέγχου του «Ansible» (Ansible control machine) προς την εικονική μηχανή με λειτουργικό σύστημα «Windows 10», ένα «module» με «ad-hoc» τρόπο, και συγκεκριμένα το «module» «win_command», το οποίο εκτελεί μια εντολή της γραμμής εντολών (command line) των «Windows», στην προκειμένη περίπτωση την εντολή «ipconfig», που εμφανίζει τις λεπτομέρειες του δικτύου της εικονικής μηχανής με λειτουργικό σύστημα «Windows 10», με δικαιώματα τοπικού διαχειριστή και ζητείται και ο κωδικός αυτού του διαχειριστή.

Στην παρούσα μελέτη, τα διαπιστευτήρια του λογαριασμού αυτού καταχωρήθηκαν στο αρχείο «inventory», μιας και δεν υφίσταται μεγάλος αριθμός υπολογιστών (hosts) προς παραμετροποίηση. Το αρχείο «inventory» κρυπτογραφήθηκε με τη χρήση του «Ansible

Vault», του οποίου η χρησιμότητα και ο τρόπος λειτουργίας, θα επεξηγηθούν στη συνέχεια.

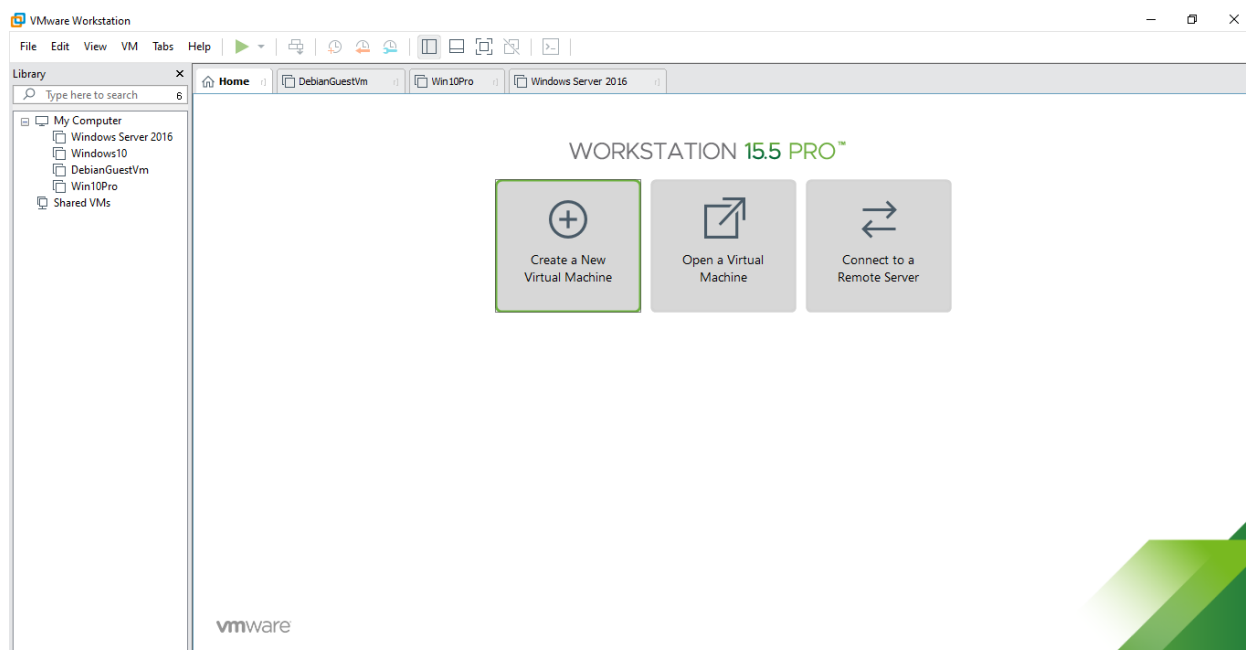
7. Για τη δημιουργία των «playbooks», σε όλες τις μελέτες περίπτωσης, χρησιμοποιήθηκε ο κειμενογράφος «Vim». Το όνομα «Vim» είναι ακρωνύμιο του «Vi iMproved», επειδή η εφαρμογή δημιουργήθηκε ως μία επέκταση του επεξεργαστή κειμένου «Vi» με πολύ περισσότερες λειτουργίες, έτσι ώστε να χρησιμεύει και στην επεξεργασία πηγαιού κώδικα εφαρμογών. Μπορεί να χρησιμοποιηθεί σε διάφορα λειτουργικά συστήματα όπως «Linux», «Unix» και «Windows». Διαθέτει γενική άδεια δημόσιας χρήσης «GPL» (General public License). Επειδή ο κειμενογράφος «Vim» βασίζεται στον κειμενογράφο «Vi», όπως προαναφέρθηκε, η διεπαφή του στηρίζεται σε περιβάλλον επικοινωνίας τερματικού (terminal) και κατά συνέπεια δεν διαθέτει κάποιο μενού, ή κάποια εικονίδια για την αλληλεπίδραση με το χρήστη. Ωστόσο, διατίθεται γραφικό περιβάλλον (GUI – Graphic User Interface) μέσω του κειμενογράφου «GVim», ο οποίος χρησιμοποιεί μενού και γραμμές εργαλείων, καθώς και όλες τις δυνατότητες πληκτρολόγησης εντολών. Τέλος, ο κειμενογράφος «Vim» διαθέτει και εγχειρίδιο για αρχάριους, καθώς και ενσωματωμένη βοήθεια σχετικά με τις εντολές και τις δυνατότητες του, η οποία είναι προσβάσιμη είτε μέσα από τον ίδιο τον «Vim», εάν έχουν εγκατασταθεί τα σχετικά αρχεία, είτε μέσω διαδικτύου⁵⁰.

3.2.1 Περιβάλλον Εργασίας του λογισμικού «VMware»

Εφόσον πραγματοποιήθηκε η εγκατάσταση του λογισμικού «VMware Workstation» στο φυσικό μηχάνημα, στη συνέχεια ακολουθήθηκαν τα παρακάτω βήματα για τη δημιουργία των εικονικών μηχανών (VMs).

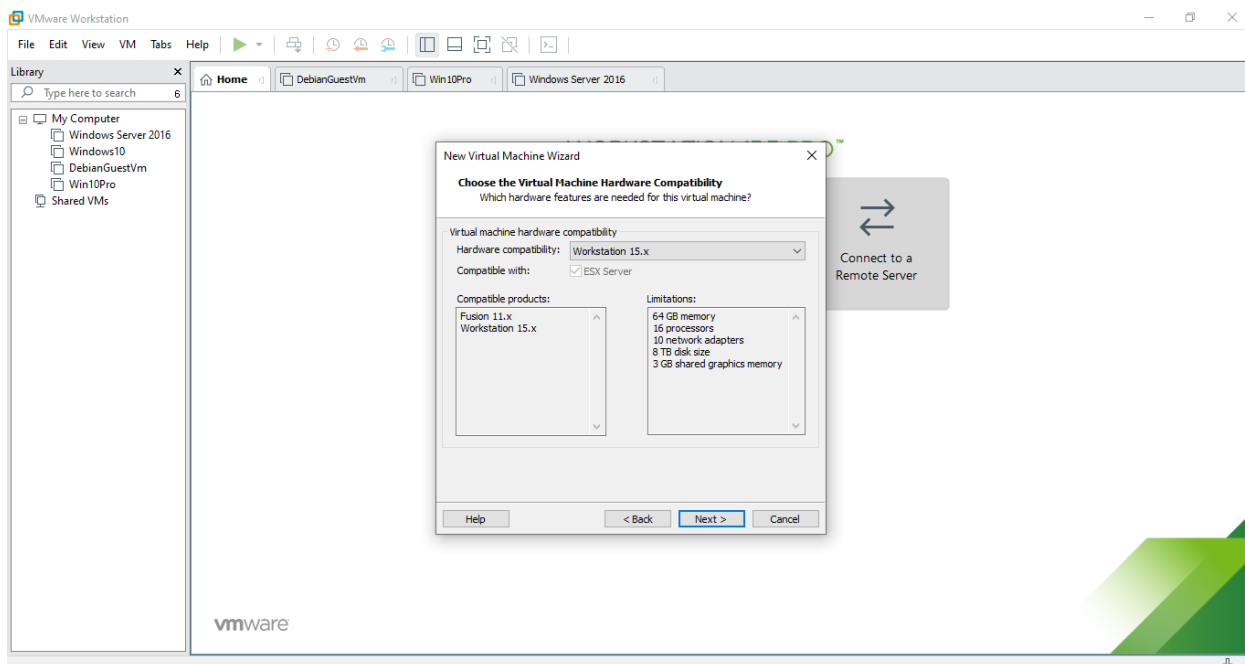
⁵⁰ <https://el.wikipedia.org/wiki/Vim>: <http://www.vim.org/>

1. Αρχικά, αφού εκκινήσει το περιβάλλον του «VMware Workstation», επιλέγεται η πρώτη επιλογή, «Create a New Virtual Machine», όπως φαίνεται και στην *Εικόνα 16*, και ανοίγει ένα νέο παράθυρο.



Εικόνα 16: Περιβάλλον του «VMware Workstation».

- Να σημειωθεί ότι εάν κάποιο ή όλα τα εικονικά μηχανήματα είχαν εξαχθεί από κάπου αλλού, από κάποιο άλλο φυσικό μηχάνημα, λόγου χάρη, θα επιλεγόταν η δυνατότητα «Open a Virtual Machine», και αν όλα όσα χρειαζόταν βρισκόταν σε κάποιον απομακρυσμένο εξυπηρετητή (server), ο οποίος φιλοξενούταν σε κάποιο περιβάλλον «VMware ESX», ή σε κάποιο άλλο «Workstation» το οποίο εκτελούταν σε κάποιο άλλο φυσικό μηχάνημα, ή σε κάποιο «VMware Vcenter Server», θα ήταν χρήσιμη η δυνατότητα «Connect to a Remote Server» .
2. Στη συνέχεια, επιλέχθηκε η προσαρμοσμένη δημιουργία της εκάστοτε εικονικής μηχανής, όπως φαίνεται και στην *Εικόνα 17*, προκειμένου να καθορισθούν οι πόροι και να λάβουν χώρα οι κατάλληλες παραμετροποιήσεις, έτσι ώστε να λειτουργούν με το βέλτιστο δυνατό τρόπο και οι τρεις (3) εικονικές μηχανές που δημιουργήθηκαν, ανάλογα με τους πόρους που δεσμεύτηκαν από το φυσικό μηχάνημα, σύμφωνα με την [παραδοχή 1 της παραγράφου 3.2.](#)



Εικόνα 17: Δημιουργία και παραμετροποίηση εικονικής μηχανής (VM) στο «VMWare Workstation».

3. Τέλος, κρίνεται απαραίτητο να σημειωθεί ότι οι τρεις (3) εικονικές μηχανές που δημιουργήθηκαν για να μπορούν να επικοινωνούν μεταξύ τους, θα πρέπει μέσω του λογισμικού «VMWare Workstation» να ρυθμιστούν ώστε να διαθέτουν δύο (2) προσαρμογείς δικτύου (network adapters). Στη συνέχεια θα πρέπει να παραμετροποιηθούν ως εξής⁵¹:

- α. Στον ένα προσαρμογέα δικτύου (Network Adapter) θα πρέπει να επιλεγεί η σύνδεση δικτύου «Bridged: Connected directly to the physical network», καθώς και να σημανθεί η επιλογή «Replicate physical network connection state».

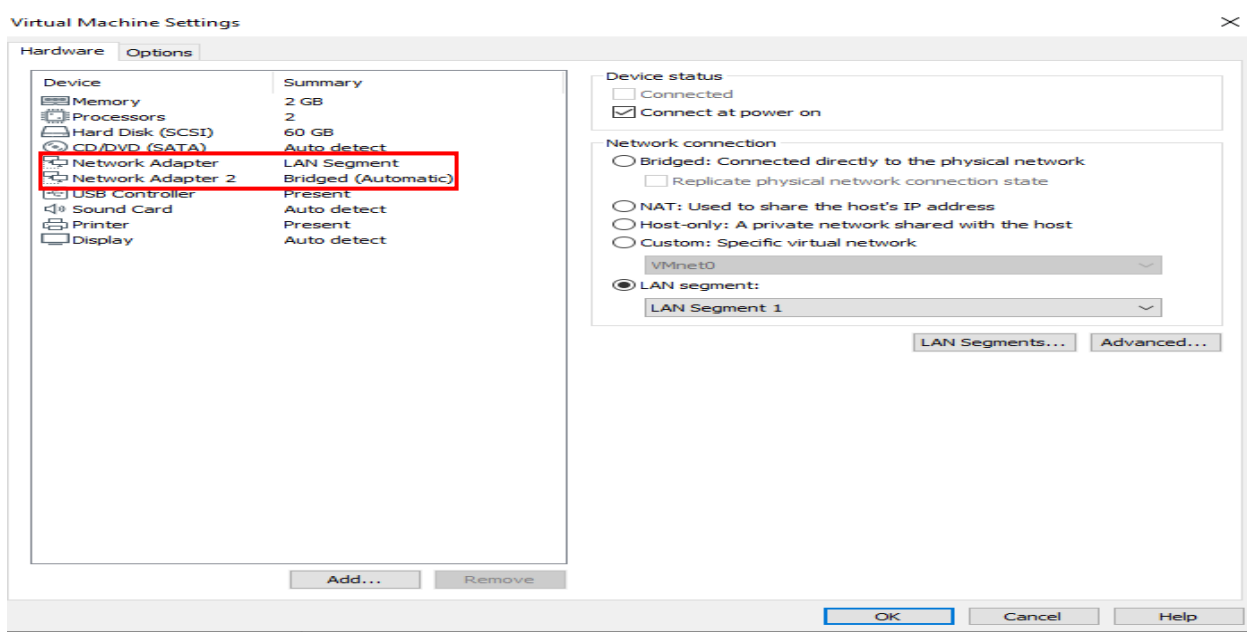
Με την επιλογή αυτή, επιτυγχάνεται η «μίμηση» της κατάστασης του φυσικού δικτύου από τα εικονικά μηχανήματα, δηλαδή το δίκτυο που έχει το φυσικό μηχάνημα, το οποίο «φιλοξενεί» το «VMWare Workstation». Συνεπώς, αν για παράδειγμα, το φυσικό μηχάνημα έχει συνδεθεί στο διαδίκτυο, τότε και οι εικονικές μηχανές που έχουν τον προσαρμογέα δικτύου ρυθμισμένη με αυτό τον τρόπο που περιγράφηκε, θα έχουν πρόσβαση στο διαδίκτυο. Για τον λόγο αυτό ρυθμίστηκε ο ένας προσαρμογέας δικτύου

⁵¹ https://www.vmware.com/support/ws5/doc/ws_team_lan_requirements.html

σε κάθε εικονικό μηχάνημα κατ' αυτόν τον τρόπο, προκειμένου να υπάρχει πρόσβαση στο «internet». Στο συγκεκριμέν προσαρμογέα δικτύου, δεν απαιτείται κάποια επιπλέον ρύθμιση.

β. Στο δεύτερο προσαρμογέα δικτύου (Network Adapter) θα πρέπει να επιλεγεί η σύνδεση «LAN Segment».

Όπως ένας φυσικός υπολογιστής πρέπει να διαθέτει προσαρμογέα δικτύου για κάθε φυσική σύνδεση δικτύου, ομοίως, μια εικονική μηχανή πρέπει να ρυθμιστεί με έναν προσαρμογέα εικονικού δικτύου για κάθε τμήμα «LAN» (LAN Segment) με το οποίο αλληλεπιδρά. Για να συνδεθεί μια εικονική μηχανή σε πολλά τμήματα «LAN» ταυτόχρονα, πρέπει να ρυθμιστεί αυτή η εικονική μηχανή με πολλούς προσαρμογείς δικτύου. Με την επιλογή αυτή, επιτυγχάνεται η δημιουργία, ενός εικονικού «τοπικού» δικτύου μεταξύ των εικονικών μηχανών.



Εικόνα 18: Παραμετροποίηση προσαρμογέων δικτύου (Network Adapters)

Για να επιτευχθεί αυτό, δημιουργήθηκε στον συγκεκριμένο προσαρμογέα δικτύου ένα τμήμα «LAN» (LAN Segment), το οποίο ονομάστηκε «LAN Segment 1». Μέσα από το «LAN Segment» που δημιουργήθηκε, οι τρεις (3) εικονικές μηχανές που δημιουργήθηκαν, επικοινωνούν μεταξύ τους, με τον εξής τρόπο:

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

➤ **Διευθύνσεις «IP» τμήματος «LAN» (LAN segment).**

Κάθε υπολογιστής - πελάτης δικτύου πρέπει να έχει «διεύθυνση IP» για δικτύωση σύμφωνα με το πρωτόκολλο «TCP / IP». Σε αντίθεση με την επιλογή δικτύωσης στο «VMware Workstation» μόνο για υπολογιστές (host only network) και την επιλογή δικτύωσης «NAT», τα τμήματα «LAN» (LAN segments), δεν διαθέτουν εξυπηρετητή «DHCP» που παρέχεται αυτόματα από το «VMware Workstation». Επομένως, πρέπει η «διεύθυνση IP» να ρυθμιστεί με μη αυτοματοποιημένο τρόπο για τις εικονικές μηχανές ομάδας σε ένα τμήμα «LAN». Για τον σκοπό αυτό, υπάρχουν δύο επιλογές:

- Η πρώτη επιλογή αφορά στη χρήση «DHCP». Για να επιτευχθεί αυτό, θα πρέπει να διαμορφωθεί ένας εξυπηρετητής που θα διαθέτει το ρόλο «DHCP» (DHCP Server) στο «LAN Segment», ο οποίος θα εκχωρεί «διευθύνσεις IP» στις εικονικές μηχανές μέσα στο ίδιο «LAN Segment».
- Η δεύτερη επιλογή αφορά σε στατικές «IP διευθύνσεις». Στην παρούσα μελέτη, αρχικά εφαρμόστηκε η συγκεκριμένη μέθοδος. Δόθηκαν, δηλαδή στις εικονικές μηχανές, στατικές «IP διευθύνσεις».
- Στις εικονικές μηχανές με λειτουργικό σύστημα «Windows» το δίκτυο στο «LAN Segment» ορίστηκε μέσα από το «Windows» περιβάλλον σε ιδιωτικό (Private).

Αποτέλεσμα της παραπάνω πρακτικής, αποτελεί η δημιουργία τριών (3) εικονικών μηχανών με τεχνικά χαρακτηριστικά, τα οποία καλύπτουν τις ελάχιστες απαιτήσεις πόρων και την εύρυθμη λειτουργία και «συνεργασία» των εικονικών μηχανών με τον φυσικό υπολογιστή που τις φιλοξενεί μέσω του «VMWare Workstation». Επιπλέον, τα δικτυακά χαρακτηριστικά καλύπτουν τις απαιτήσεις δημιουργίας ενός εικονικού τοπικού δικτύου μεταξύ των εικονικών μηχανών, καθώς επίσης και την πρόσβαση αυτών στο διαδίκτυο. Συγκεκριμένα:

▪ **Windows Server 2016**

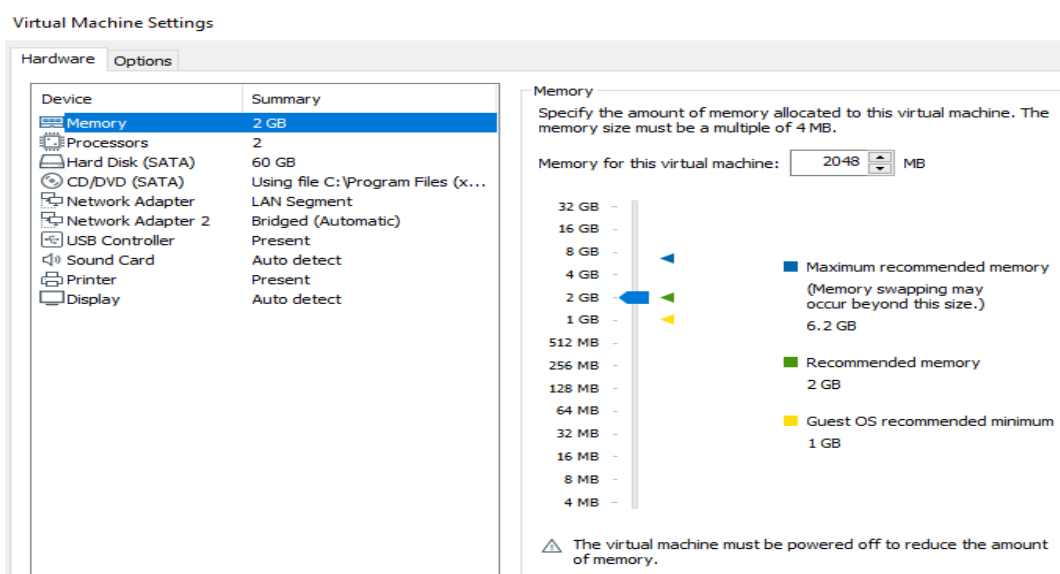
“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Η εικονική μηχανή με λειτουργικό σύστημα «Windows Server 2016» ονομάστηκε «Windows Server 2016». Ο ένας προσαρμογέας δικτύου «μιμείται» (replicate) το δίκτυο του φυσικού υπολογιστή και συνδέεται στο διαδίκτυο (Internet), και ο άλλος που βρίσκεται το «LAN Segment» έχει λάβει τις εξής τιμές:

IP: 192.168.162.1

Subnet Mask: 255.255.255.0



Εικόνα 19: Χαρακτηριστικά «Windows Server 2016 VM».

▪ Windows 10

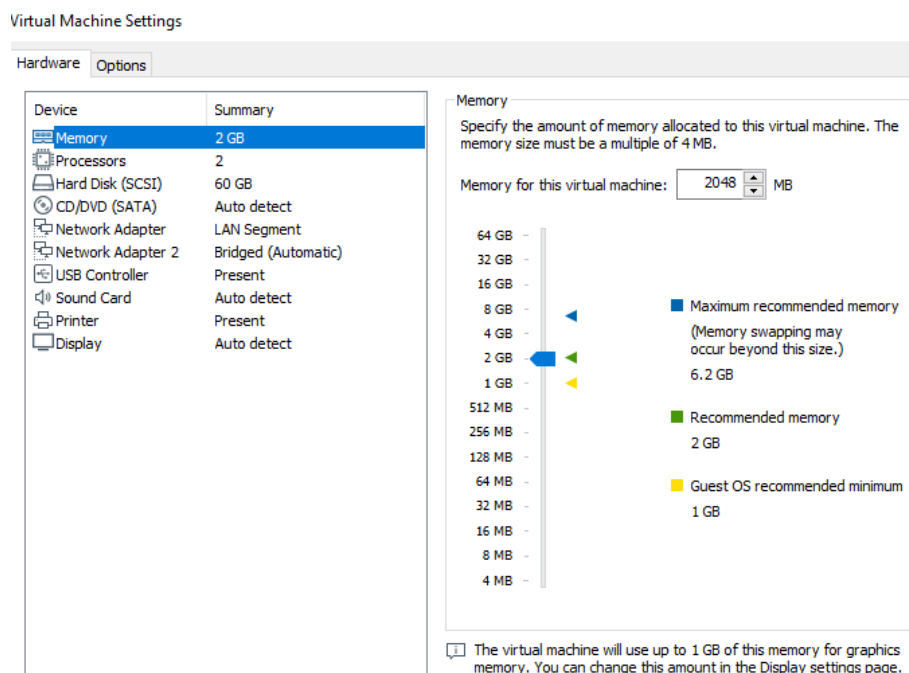
Η εικονική μηχανή με λειτουργικό σύστημα «Windows 10» ονομάστηκε «Win10Pro». Ο ένας προσαρμογέας δικτύου «μιμείται» (replicate) το δίκτυο του φυσικού υπολογιστή και συνδέεται στο διαδίκτυο (Internet), και ο άλλος που βρίσκεται το «LAN Segment» έχει λάβει τις εξής τιμές:

IP: 192.168.162.3

Subnet Mask: 255.255.255.0

Gateway: 192.168.162.2 (Η «IP διεύθυνση» του «Debian 10 Vm»)

DNS: 192.168.162.1 (Η «IP διεύθυνση» του «Windows Server 2016»)



Εικόνα 20: Χαρακτηριστικά «Windows 10 Professional VM».

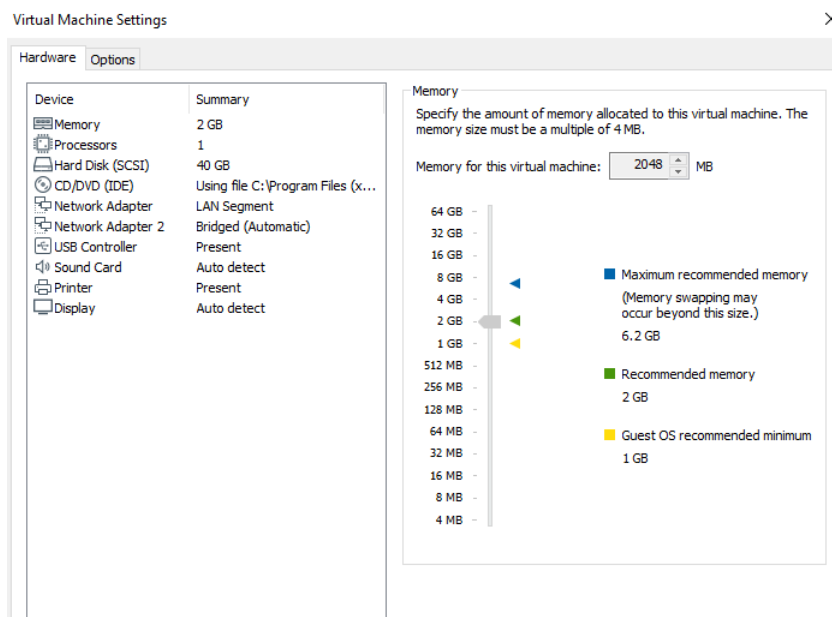
■ Debian 10 (Buster)

Η εικονική μηχανή με λειτουργικό σύστημα «Linux Debian 10 Buster» ονομάστηκε «DebianGuestVm». Ο ένας προσαρμογέας δικτύου «μιμείται» (replicate) το δίκτυο του φυσικού υπολογιστή και συνδέεται στο διαδίκτυο (Internet), και ο άλλος που βρίσκεται το «LAN Segment» έχει λάβει τις εξής τιμές:

IP: 192.168.162.2

Subnet Mask: 255.255.255.0

Gateway & DNS: 192.168.162.1 (Η «IP διεύθυνση» του «Windows Server 2016»)



Εικόνα 21: Χαρακτηριστικά «Linux Debian 10 Buster VM».

3.2.2 Εικονική μηχανή «Linux – Debian 10 Buster»

Για τη δημιουργία του μηχανήματος ελέγχου του «Ansible» (Ansible control machine), επιλέχθηκε η διανομή «Linux Debian 10 – Buster», η οποία εγκαταστάθηκε σε εικονικό περιβάλλον, όπως προαναφέρθηκε. Η επιλογή της συγκεκριμένης διανομής έγινε, διότι αποτελεί μια διανομή αρκετά οικεία προς τον χρήστη και σε κάποιον ο οποίος δεν έχει ιδιαίτερη εξοικείωση με τα περιβάλλοντα «Linux». Επιπλέον, η συγκεκριμένη διανομή είναι «συμβατή» για χρήση σε συνδυασμό με το «Ansible», γεγονός που ενίσχυσε την απόφαση για τελική επιλογή της εν λόγω διανομής⁵².

Κρίνεται σκόπιμο να αναφερθεί ότι το «Ansible», ναι μεν μπορεί να χρησιμοποιηθεί για την αυτοματοποιημένη παραμετροποίηση λειτουργικών συστημάτων «Windows», πλην όμως, δεν είναι εφικτό να εγκατασταθεί σε περιβάλλοντα «Windows»⁵³. Μολονότι, δεν αποτελεί ορθή πρακτική, καθόσον το «Ansible» δεν υποστηρίζει το υποσύστημα «Linux» των «Windows»

⁵² https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#selecting-an-ansible-version-to-install

⁵³ <https://www.ansible.com/integrations/infrastructure/windows>

(Windows Subsystem Linux – WSL), μπορεί να εγκατασταθεί σε αυτό. Για το λόγο αυτό δεν συνιστάται η χρήση της μεθόδου αυτής σε παραγωγικά περιβάλλοντα⁵⁴.

3.2.2.1 Εγκατάσταση του εργαλείου «Ansible»

Αρχικά, θα πρέπει να υπάρχει στην εικονική μηχανή με την ονομασία «DebianGuestVm» ένας χρήστης που να έχει δικαιώματα υπερχρήστη (superuser), ώστε να μπορεί να εκτελεί εντολές με διαχειριστικά δικαιώματα (super user do – sudo). Για το σκοπό αυτό, στην ανωτέρω αναφερόμενη εικονική μηχανή έχει δημιουργηθεί ένας τοπικός χρήστης με το όνομα «debuseradmin»⁵⁵.

Προκειμένου να εγκατασταθεί το «Ansible», εκτελέστηκαν οι ακόλουθες εντολές που περιγράφονται στη συνέχεια, από τη γραμμή εντολών (terminal).

Αρχικά, έγινε λήψη των τελευταίων αναβαθμίσεων του λειτουργικού συστήματος «Debian 10» με τη χρήση της εντολής:

```
sudo apt update
```

Τα πακέτα του «Ansible» είναι διαθέσιμα σε προεπιλεγμένα αποθετήρια για τη διανομή «Debian 10». Συνεπώς, προκειμένου αυτό το αποθετήριο να προστεθεί στο σύστημα, εκτελέστηκε η εντολή⁵⁶:

⁵⁴https://docs.ansible.com/ansible/latest/user_guide/windows_faq.html.
<https://books.google.gr/books?id=qvnpDwAAQBAJ&pg=PA9&lpg=PA9&dq=WSL+Ansible&source=bl&ots=RMM3NekMqM&sig=ACfU3U1L7mwy-guZPA5I76IPq6NxBJ9fbw&hl=en&sa=X&ved=2ahUKEwjghdrw4KzqAhWCGewKHSnaApUQ6AEwCXoECAgQAQ#v=onepage&q=WSL%20Ansible&f=false>

⁵⁵ https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#selecting-an-ansible-version-to-install.
<https://www.linuxtechi.com/install-ansible-automation-tool-debian10/>.
https://linuxhint.com/install_ansible_debian10/

⁵⁶ <https://launchpad.net/~ansible/+archive/ubuntu/ansible>

```
sudo add-apt-repository ppa:ansible/ansible
```

Στη συνέχεια, πραγματοποιήθηκε η εγκατάσταση του «Ansible» με τη χρήση της εντολής:

```
sudo apt install ansible
```

Τέλος, με την ολοκλήρωση της εκτέλεσης των προαναφερόμενων εντολών, εκτελέστηκε η παρακάτω εντολή, προκειμένου να διαπιστωθεί η ύπαρξη και η έκδοση του «Ansible» στο σύστημα:

```
ansible --version
```

Το αποτέλεσμα έχει ως εξής:

```
deuser@ubuntu:~$ ansible --version
ansible 2.9.9
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/deuseradmin/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.16 (default, Oct 10 2019, 22:02:15) [GCC 8.3.0]
```

Εικόνα 22: Επιτυχής εγκατάσταση του «Ansible».

Επειδή οι εξαρτήσεις της βιβλιοθήκης «pywinrm», οι οποίες είναι απαραίτητες για τη χρήση της υπηρεσίας «WinRM», δεν εμπεριέχονται στο «Ansible», θα πρέπει να πραγματοποιηθεί η εγκατάσταση της στην εικονική μηχανή που είναι εγκατεστημένο το «Ansible»⁵⁷. Η απλούστερη μέθοδος είναι η εκτέλεση της εντολής:

```
pip install pywinrm
```

Έπειτα, πραγματοποιήθηκε εισαγωγή της βιβλιοθήκης «pywinrm», χρησιμοποιώντας τη γλώσσα προγραμματισμού «python» όπως παρακάτω:

⁵⁷https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html
<https://www.ansible.com/blog/connecting-to-a-windows-host>

```
>python  
>import winrm
```

3.2.2.2 Δημιουργία και παραμετροποίηση του αρχείου «inventory» του «Ansible»

Το «Ansible» χρησιμοποιείται για την ταυτόχρονη αυτοματοποιημένη παραμετροποίηση πολλαπλών διαχειριζόμενων κόμβων στην υποδομή πληροφορικής ενός παραγωγικού περιβάλλοντος, χρησιμοποιώντας μια λίστα ή μια ομάδα λιστών που γνωρίζει ως «ευρετήριο» (inventory). Μόλις καθοριστεί το αρχείο «inventory», μπορούν να χρησιμοποιηθούν μοτίβα για την επιλογή των κεντρικών υπολογιστών ή των ομάδων στις οποίες θα εκτελεστεί το «Ansible». Συνεπώς, το αρχείο «inventory» αποτελεί το αρχείο ή/και τα αρχεία που περιλαμβάνουν τους υπολογιστές – στόχους του «Ansible», καθώς και τις όποιες μεταβλητές τους αφορούν⁵⁸.

Η προεπιλεγμένη θέση για το αρχείο «inventory» είναι ένα αρχείο που ονομάζεται «hosts» και βρίσκεται στην διαδρομή «/etc/ansible/hosts». Ωστόσο, δεν παρεμποδίζεται ο καθορισμός ενός νέου, διαφορετικού αρχείου «inventory». Το κάθε αρχείο «inventory» μπορεί να κληθεί στη γραμμή εντολών χρησιμοποιώντας την επιλογή «-i <path>», όπου «path» είναι η διαδρομή του εκάστοτε αρχείου «inventory». Επίσης, είναι εφικτό να χρησιμοποιηθούν πολλά αρχεία «inventory» ταυτόχρονα ή / και να κληθούν αρχεία «inventory» από δυναμικές πηγές ή πηγές «cloud» ή διαφορετικές μορφές (YAML, ini κ.λπ.).

Το αρχείο «inventory» μπορεί να έχει μία από τις πολλές υπάρχουσες μορφές με πιο συνηθισμένες τις «INI» και «YAML».

Ένα τυπικό αρχείο «inventory» τύπου «INI» έχει την εξής μορφή:

```
mail.example.com  
  
[webserver]  
foo.example.com  
bar.example.com
```

⁵⁸ https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html

```
[dbservers]
one.example.com
two.example.com
three.example.com
```

Ένα τυπικό αρχείο «inventory» τύπου «YAML» έχει την εξής μορφή:

```
all:
  hosts:
    mail.example.com:
  children:
    webservers:
      hosts:
        foo.example.com:
        bar.example.com:
    dbservers:
      hosts:
        one.example.com:
        two.example.com:
        three.example.com:
```

Για τη δημιουργία του αρχείου «inventory», υπάρχουν δύο (2) προεπιλεγμένες ομάδες: «όλες» (all) και «μη ομαδοποιημένες» (ungrouped). Η ομάδα «all» περιέχει κάθε κεντρικό υπολογιστή. Η ομάδα «ungrouped» περιέχει όλους τους κεντρικούς υπολογιστές που δεν έχουν άλλη ομάδα εκτός από την ομάδα «all». Κάθε υπολογιστής ανήκει πάντα σε τουλάχιστον δύο (2) ομάδες («all» και «ungrouped» ή «all» και κάποια άλλη ομάδα).

Ένας υπολογιστής (host) μπορεί να ανήκει σε περισσότερες από μία ομάδες, κι αυτό είναι το πιθανότερο να συμβαίνει στην υποδομή. Επιπλέον, στο αρχείο «inventory» μπορούν να αποθηκευθούν μεταβλητές με τιμές που σχετίζονται με ένα συγκεκριμένο υπολογιστή ή ομάδα στο αρχείο «inventory». Οι μεταβλητές αυτές μπορούν να προστεθούν στο κύριο αρχείο

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

«inventory». Καθώς, όμως, προστίθενται όλο και περισσότεροι κόμβοι προς διαχείριση στο αρχείο «inventory», θα ήταν προτιμότερη η αποθήκευση των μεταβλητών σε ξεχωριστά αρχεία.

Οι μεταβλητές ομάδας είναι ένας βολικός τρόπος για την εφαρμογή μεταβλητών σε πολλούς υπολογιστές ταυτόχρονα. Πριν από την εκτέλεση, ωστόσο, το «Ansible» ισοσταθμίζει πάντα τις μεταβλητές, συμπεριλαμβανομένων των μεταβλητών του αρχείου «inventory», στο επίπεδο του υπολογιστή. Εάν ένας υπολογιστής είναι μέλος πολλών ομάδων, το «Ansible» διαβάζει μεταβλητές με τιμές από όλες αυτές τις ομάδες. Εάν εκχωρηθούν διαφορετικές τιμές στην ίδια μεταβλητή σε διαφορετικές ομάδες, τότε το «Ansible» επιλέγει ποια τιμή θα χρησιμοποιηθεί βάσει εσωτερικών κανόνων για συγχώνευση.

Το αρχείο «inventory» που δημιουργήθηκε για την εκπόνηση της εν λόγω διπλωματικής εργασίας, είναι τύπου «INI», ονομάστηκε «inventory.ini» και δημιουργήθηκε στη διαδρομή «/home/debuseradmin/inventory.ini». Σε αυτό το αρχείο, προστέθηκαν οι μεταβλητές που χρησιμοποιεί το «Ansible», προκειμένου να συνδεθεί και να επικοινωνεί με την εικονική μηχανή που ονομάστηκε «Win10Pro». Το αρχείο αυτό έχει της εξής μορφή:

```
[winhost]
192.168.162.3

[winhost:vars]
ansible_connection= winrm
ansible_winrm_transport= ntlm
ansible_user= admin
ansible_password= "{{ vault_ansible_password }}"
ansible_winrm_server_cert_validation= ignore
ansible_port= 5986
```

winhost: Το όνομα του κεντρικού υπολογιστή για σύνδεση, ή η «IP διεύθυνσή» του. Στην προκειμένη περίπτωση είναι η «IP» του.

ansible connection: Αποτελεί τον τύπο σύνδεσης με τις εικονικές μηχανές που έχουν λειτουργικό σύστημα έκδοσης «Windows». Στην εκπόνηση της εν λόγω διπλωματικής μελέτης, επιλέχθηκε η υπηρεσία «WinRM», καθώς μελετάται η αυτοματοποιημένη παραμετροποίηση υπολογιστή – στόχου με λειτουργικό σύστημα «Windows» (Windows host).

ansible port: Είναι ο αριθμός της θύρας σύνδεσης. Στη συγκεκριμένη περίπτωση είναι ο αριθμός «5986», η οποία είναι η προεπιλεγμένη θύρα στα λειτουργικά συστήματα «Windows» και χρησιμοποιείται από τον «ακροατή WinRM» (WinRM listener), μέσω του πρωτοκόλλου «HTTPS».

ansible user: Είναι το όνομα του χρήστη που χρησιμοποιείται κατά τη σύνδεση στην εικονική μηχανή με λειτουργικό σύστημα «Windows». Στη συγκεκριμένη περίπτωση, είναι ο χρήστης, που ονομάζεται «admin» και έχει δικαιώματα τοπικού διαχειριστή (local administrator), προκειμένου μέσω αυτού του χρήστη να μπορούν να εκτελεστούν οι εκάστοτε εντολές ή «playbooks» στην εν λόγω εικονική μηχανή. Ο συγκεκριμένος χρήστης θα μπορούσε να ορισθεί και «ad hoc» κατά την εκτέλεση της οποιαδήποτε εντολής ή «playbook» από την εικονική μηχανή που περιέχει το «Ansible» με τη χρήση της παραμέτρου «-u [username]».

ansible password: Είναι ο κωδικός πρόσβασης του προαναφερόμενου χρήστη, ο οποίος θα χρησιμοποιηθεί για την αυθεντικοποίηση με την εικονική μηχανή που ονομάζεται «Win10Pro». Η τιμή της μεταβλητής αυτής αποτελεί «ευαίσθητο» δεδομένο και κρυπτογραφήθηκε με τη χρήση του «Ansible Vault», όπως θα αναλυθεί στην παράγραφο 3.2.2.3 του παρόντος Κεφαλαίου.

3.2.2.3 Κρυπτογράφηση με τη χρήση του χαρακτηριστικού «Ansible Vault»

Το αρχείο «inventory» που δημιουργήθηκε παραπάνω περιέχει αρκετά ευαίσθητα δεδομένα, για την προστασία των οποίων χρησιμοποιήθηκε το «Ansible Vault». Το «Vault» είναι ένα χαρακτηριστικό του «Ansible» που επιτρέπει τη διατήρηση «ευαίσθητων» δεδομένων, όπως κωδικούς πρόσβασης κλπ με δυνατότητα κρυπτογράφησης και όχι ως απλό κείμενο σε κάποιο αρχείο, το οποίο μπορεί να προσπελαστεί και να αποκτηθεί η πρόσβαση στα ευαίσθητα αυτά

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

δεδομένα από μη εξουσιοδοτημένο προσωπικό. Υλοποιείται με λεπτομέρεια σε επίπεδο αρχείου, που σημαίνει ότι τα αρχεία είναι είτε πλήρως κρυπτογραφημένα είτε μη κρυπτογραφημένα. Για την κρυπτογράφηση των δεδομένων το «Ansible» χρησιμοποιεί τον κρυπταλγόριθμο «AES» (256-bit). Ο «AES-256» παρέχει συμμετρική κρυπτογράφηση με κωδικό πρόσβασης που παρέχεται από τον χρήστη. Αυτό σημαίνει ότι ο ίδιος κωδικός χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση του περιεχομένου. Συνεπώς, για να θεωρηθεί ένα αρχείο στο «Ansible Vault» ασφαλές, και να μην είναι δυνατή η αποκρυπτογράφηση του, θα πρέπει να δοθεί κάποιος όσο γίνεται πιο τυχαίος και «δύσκολος» κωδικός είναι δυνατόν⁵⁹. Το «Ansible» είναι σε θέση να εντοπίσει και να αποκρυπτογραφήσει τυχόν κρυπτογραφημένα αρχεία ή μεταβλητές, τα οποία κρυπτογραφήθηκαν με το «Ansible Vault», κατά την εκτέλεση ενός «playbook» ή ενός «task» σε ένα «playbook»⁶⁰.

Υπάρχουν δύο (2) περιεχόμενα τύπου «Vault», με το καθένα να έχει τις δικές του χρήσεις και περιορισμούς:

Vault αρχεία: Το πλήρες αρχείο είναι κρυπτογραφημένο στο «θησαυροφυλάκιο» (Vault) και μπορεί να περιέχει μεταβλητές του «Ansible» ή οποιοδήποτε άλλο είδος περιεχομένου. Αποκρυπτογραφείται μόνο όταν «φορτώνεται» ή γίνεται αναφορά σε αυτό, κι αυτό μόνο κατόπιν χρήσης του κωδικού πρόσβασης που ορίστηκε. Αυτό συμβαίνει διότι το «Ansible» δεν μπορεί να γνωρίζει εάν χρειάζεται το περιεχόμενο, εκτός εάν το αποκρυπτογραφήσει. Μπορεί να χρησιμοποιηθεί σε αρχεία «inventory», καθώς επίσης και σε οποιοδήποτε αρχείο περιέχει και «φορτώνει» μεταβλητές (π.χ. «vars_files», «group_vars», «host_vars», «include_vars», κλπ) και για ορισμένες ενέργειες που αφορούν σε αρχεία.

Μονή κρυπτογραφημένη μεταβλητή: Σε αυτή την περίπτωση, μόνο συγκεκριμένες μεταβλητές κρυπτογραφούνται μέσα σε ένα κανονικό «αρχείο μεταβλητής» (var_file). Δεν λειτουργεί για άλλο περιεχόμενο, παρά μόνο για μεταβλητές. Γίνεται αποκρυπτογράφηση κατ' απαίτηση, ώστε να είναι εφικτή η ύπαρξη και διατήρηση μεταβλητών σε διάφορα «μυστικά» αρχεία. Επιπλέον,

⁵⁹https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#variables-and-vaults
https://docs.ansible.com/ansible/latest/user_guide/playbooks_vault.html

⁶⁰<https://www.digitalocean.com/community/tutorials/how-to-use-vault-to-protect-sensitive-ansible-data-on-ubuntu-16-04>

είναι δυνατή η συνύπαρξη κρυπτογραφημένων και μη κρυπτογραφημένων μεταβλητών στο ίδιο αρχείο, ακόμα και ενσωματωμένες σε ένα «Ansible playbook» ή έναν ρόλο του «Ansible» (Ansible role).

Το αρχείο «inventory.ini», καθώς επίσης και τα «playbooks» που δημιουργήθηκαν και εξετάζονται στη συνέχεια, περιέχουν μεταβλητές με τιμές κωδικούς πρόσβασης, που τις καθιστά «ευαίσθητες» μεταβλητές. Προκειμένου να υλοποιηθεί η διαδικασία κρυπτογράφησης των μεταβλητών αυτών ακολουθήθηκε η εξής διαδικασία:

Αρχικά δημιουργήθηκαν τα αρχεία «vars.yml» και «vault.yml» στη διαδρομή «/home/debuseradmin/group_vars/winhost». Τα αρχεία δημιουργήθηκαν εκτελώντας τις εντολές:

```
vim /home/debuseradmin/group_vars/winhost/vars.yml
```

```
vim /home/debuseradmin/group_vars/winhost/vault.yml
```

Συνεπώς, προκύπτει η δομή φακέλων και αρχείων ή «δέντρο» (tree) όπως φαίνεται εκτελώντας την παρακάτω εντολή:

```
tree /home/debuseradmin/group_vars
```

```
debuseradmin@debian:~$ tree /home/debuseradmin/group_vars/  
/home/debuseradmin/group_vars/  
├── winhost  
│   ├── vars.yml  
│   └── vault.yml  
  
1 directory, 2 files
```

Εικόνα 23: Δομή «Ansible Vault».

Στο αρχείο «vault.yml», τοποθετήθηκαν οι τιμές των μεταβλητών με τα πραγματικά τους δεδομένα, διότι το αρχείο στη συνέχεια θα κρυπτογραφηθεί. Επιπλέον, κάθε μεταβλητή διαθέτει το πρόθεμα «vault_». Επομένως, αρχικά το αρχείο «vault.yml» έχει την εξής μορφή:

```
---  
  
vault_ansible_password: "κάποιος κωδικός"  
  
vault_ansible_become_password: "κάποιος κωδικός"  
  
vault_domain_admin_password: "κάποιος κωδικός"
```

Έπειτα, το αρχείο «vault.yml» κρυπτογραφήθηκε με τη χρήση της εντολής:

```
ansible-vault encrypt vault.yml
```

Εκτελώντας την παραπάνω εντολή, ζητήθηκε κωδικός για την υλοποίηση του «Ansible Vault» και κατ' επέκταση της διαδικασίας κρυπτογράφησης. Για την αποκρυπτογράφηση του εκάστοτε αρχείου, θα πρέπει αντί για την παράμετρο «encrypt», να δοθεί στην εντολή η παράμετρος «decrypt».

Στην *Εικόνα 24* εμφανίζεται το παραγόμενο κρυπτογράφημα (ciphertext).

μορφών σήμανσης που επιστρέφονται στο χρήστη μέσω αιτήματος «HTTP». Έχει τη μορφή «”{{ exampletext }}”»⁶¹.

Κατά συνέπεια και με βάση ό,τι περιγράφηκε παραπάνω, το αρχείο «vars.yml» διαθέτει την εξής μορφή:

```
---

ansible_password: "{{ vault_ansible_password }}"

ansible_become_password: "{{ vault_ansible_become_password }}"

domain_admin_password: "{{ vault_domain_admin_password }}"
```

Σημειώνεται ότι όπου απαντώνται οι παραπάνω μεταβλητές, είτε σε κάποιο «playbook», είτε σε κάποιο «task» ενός «playbook», η τιμή που ορίζεται σε αυτές είναι η ίδια ακριβώς με αυτή του αρχείου «vars.yml». Αυτό συμβαίνει προκειμένου να μπορεί το «Ansible» να εντοπίζει τις μεταβλητές αυτές και να εκτελεί με ορθό τρόπο ό,τι ζητείται. Τέλος, γίνεται μνεία πως είναι δυνατή η δημιουργία αρκετών τέτοιων αρχείων «Vault». Για τις ανάγκες της παρούσας εργασίας, δημιουργήθηκε μόνο ένα αρχείο «Ansible Vault».

⁶¹[https://en.wikipedia.org/wiki/Jinja_\(template_engine\)](https://en.wikipedia.org/wiki/Jinja_(template_engine))·
<https://codeburst.io/jinja-2-explained-in-5-minutes-88548486834e>

3.2.3 Εικονική μηχανή "Windows 10"

Στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10», προκειμένου αυτή να μπορεί να παραμετροποιηθεί με τη χρήση των εργαλείων αυτοματοποίησης που προσφέρει το «Ansible», εφαρμόζονται και χρησιμοποιούνται οι δυνατότητες που δίνει η υπηρεσία «WinRM» (Windows Remote Management). Κι αυτό, διότι τα λειτουργικά συστήματα «Windows», ενώ με κάποιο τρόπο είναι εφικτό να χρησιμοποιούν «SSH» (Secure Shell), όπως τα λειτουργικά συστήματα διανομής «Linux» ή/και «Unix», αυτό δεν προτείνεται, καθώς η χρήση του «SSH» στα περιβάλλοντα «Windows» βρίσκεται σε πειραματικό στάδιο⁶² και οι εντολές που ορίζονται προς εκτέλεση με κάθε ένα από τα «playbooks» που εκτελούνται μέσω του «Ansible» στον υπολογιστή - στόχο, χρησιμοποιεί τα «cmd-lets» της διεπαφής «Powershell»⁶³.

Προκειμένου να είναι δυνατή η χρήση του «Ansible» στα λειτουργικά συστήματα «Windows», απαιτείται η κάλυψη των εξής απαιτήσεων⁶⁴:

- Το «Ansible» μπορεί να διαχειρίζεται εκδόσεις των λειτουργικών συστημάτων «Windows», οι οποίες υποστηρίζονται από την εταιρεία «Microsoft». Το «Ansible» μπορεί να διαχειριστεί εκδόσεις επιτραπέζιων υπολογιστών (desktop) των «Windows 7, 8.1, and 10», και εκδόσεις εξυπηρετητών (Server), συμπεριλαμβανομένων των «Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, and 2019».
- Το «Ansible» απαιτεί τουλάχιστον τη διεπαφή «PowerShell 3.0» ή νεότερη έκδοση της, και τουλάχιστον το περιβάλλον λογισμικού (framework) «.NET 4.0» να είναι εγκατεστημένα στον υπολογιστή – στόχο με λειτουργικό σύστημα «Windows» (Windows host).
- Επιπλέον, απαιτείται ένας «ακροατής WinRM» (WinRM listener) να δημιουργηθεί και να ενεργοποιηθεί στον υπολογιστή – στόχο με λειτουργικό σύστημα «Windows» (Windows host).

⁶² <https://github.com/PowerShell/Win32-OpenSSH/wiki/Install-Win32-OpenSSH>

⁶³ <https://www.ansible.com/blog/connecting-to-a-windows-host>
https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html

⁶⁴ https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html

- Εάν τα παρακάτω βήματα εκτελεστούν με τη συγκεκριμένη διαδοχική σειρά, τότε ο υπολογιστής - στόχος θα έχει παραμετροποιηθεί με ορθό τρόπο και θα είναι δυνατή η χρήση του «Ansible» για περαιτέρω μελέτη και εφαρμογή περιπτώσεων αυτοματοποιημένων διαδικασιών και ενεργειών.

3.2.3.1 Εγκατάσταση – Παραμετροποίηση της υπηρεσίας «WinRM» (Windows Remote Management)

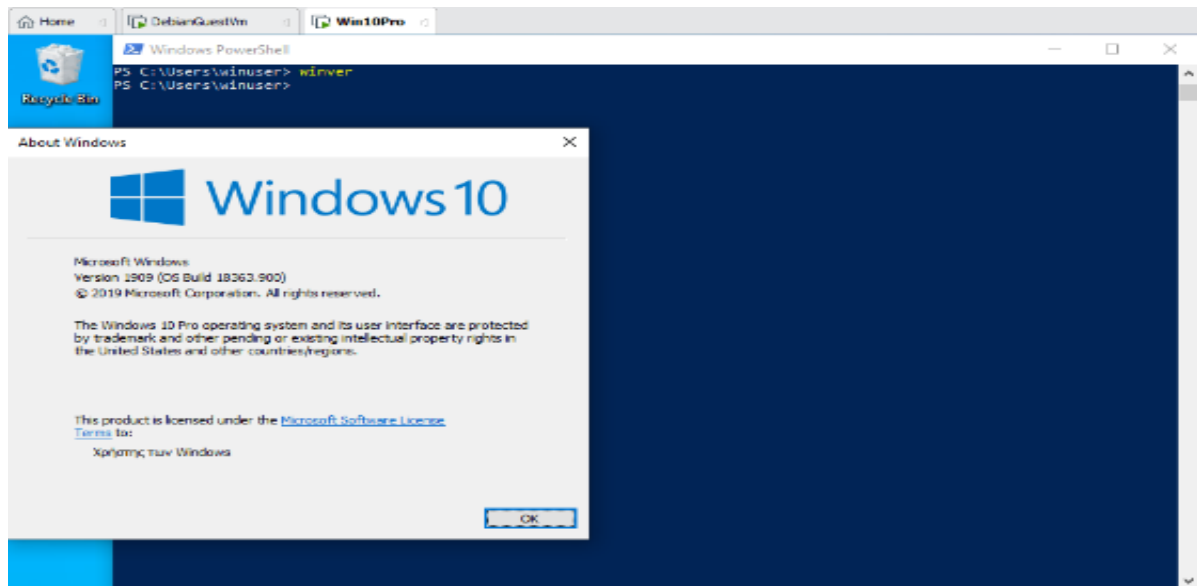
Προκειμένου να εδραιωθεί η δυνατότητα επικοινωνίας μεταξύ του υπολογιστή – στόχου με λειτουργικό σύστημα «Windows» (Windows host) και του μηχανήματος ελέγχου «Ansible» (Ansible control machine), ακολουθήθηκαν ορισμένες διαδικασίες και στις δύο (2) εικονικές μηχανές με λειτουργικό σύστημα «Windows 10 Professional» και «Windows Server 2016» αντίστοιχα, οι οποίες περιγράφονται στη συνέχεια⁶⁵. Να σημειωθεί ότι οι εν λόγω διαδικασίες εκτελούνται στην εικονική μηχανή με ονομασία «Win10Pro».

1. Αρχικά, ελέγχθηκε η έκδοση του λειτουργικού συστήματος των «Windows», προκειμένου να αποφασισθεί εάν είναι απαραίτητη η αναβάθμιση της έκδοσης του λειτουργικού συστήματος ή όχι. Ο εν λόγω έλεγχος πραγματοποιήθηκε με την εκτέλεση στη διεπαφή «PowerShell» της εντολής:

Winver

⁶⁵https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html·
<https://docs.ansible.com/ansible/latest/plugins/connection/winrm.html>·
<https://adamtheautomator.com/winrm-https-ansible/>·
<https://digitalist.global/talks/winrmansible/>

Το αποτέλεσμα που προκύπτει καλύπτει τις απαιτήσεις του «Ansible» οι οποίες προαναφέρθηκαν, καθώς όπως φαίνεται και στην *Εικόνα 25*, η έκδοση του λειτουργικού συστήματος «Windows 10» είναι έκδοσης «1909» και αρχιτεκτονικής «x64».



Εικόνα 25: Έκδοση Λειτουργικού Συστήματος.

2. Έπειτα, ελέγχθηκε η έκδοση της διεπαφής «PowerShell». Αυτό πραγματοποιήθηκε με την εκτέλεση της εντολής:

`$PSVersionTable.PSVersion`

Η εντολή αυτή εμφανίζει έναν πίνακα με τα στοιχεία που αφορούν στην έκδοση της διεπαφής «PowerShell». Εάν η έκδοση της διεπαφής «PowerShell» είναι κάτω από την «3.0» κρίνεται απαραίτητη η αναβάθμισή της. Στη συγκεκριμένη μελέτη, η έκδοση της «PowerShell» είναι η «5.0», διότι χρησιμοποιείται η έκδοση «1909» των «Windows 10», όπως προαναφέρθηκε, οπότε και καλύπτονται οι απαιτήσεις του «Ansible».



```
Windows PowerShell
PS C:\Users\winuser> winver
PS C:\Users\winuser> $PSVersionTable.PSVersion

Major Minor Build Revision
-----
5      1      18362  752

PS C:\Users\winuser>
```

Εικόνα 26: Έκδοση «PowerShell».

3. Στη συνέχεια, πραγματοποιείται έλεγχος για το «PowerShell Remoting». Το «PowerShell Remoting» είναι ήδη ενεργοποιημένο στα περισσότερα σύγχρονα λειτουργικά συστήματα «Windows». Στις εκδόσεις λειτουργικού εξυπηρετητών «Windows Server» το «PowerShell Remoting» είναι ενεργοποιημένο συνήθως από προεπιλογή (default). Προκειμένου, ωστόσο, να ελεγχθεί εάν αυτό ισχύει, εκτελέστηκε στη διεπαφή «PowerShell» με δικαιώματα διαχειριστή, η εντολή⁶⁶:

Enable-PSRemoting

Το «cmdlet Enable-PSRemoting» ρυθμίζει τον υπολογιστή ώστε να λαμβάνει απομακρυσμένες εντολές μέσω της διεπαφής «PowerShell» που αποστέλλονται χρησιμοποιώντας την τεχνολογία «WS-Management». Η απομακρυσμένη χρήση της «PowerShell» με βάση την τεχνολογία «WS-Management» υποστηρίζεται προς το παρόν μόνο σε λειτουργικά συστήματα «Windows» και λειτουργεί μόνο σε ιδιωτικά δίκτυα. Για το λόγο αυτό, το δίκτυο ορίστηκε ως ιδιωτικό (private network) και στις δύο (2) εικονικές μηχανές με λειτουργικό σύστημα «Windows 10» και «Windows Server 2016» αντίστοιχα.

Η εντολή «Enable-PSRemoting» εκτελεί τις ακόλουθες λειτουργίες:

- Εκτελεί το «cmdlet Set-WSManQuickConfig», το οποίο στη συνέχεια εκτελεί τις ακόλουθες εργασίες:

⁶⁶<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

- ❖ Εκκινεί την υπηρεσία «WinRM».
- ❖ Ορίζει τον τύπο εκκίνησης στην υπηρεσία «WinRM» σε «Αυτόματο» (Automatic).
- ❖ Δημιουργεί έναν «ακροατή» (listener), έτσι ώστε να γίνονται δεκτά αιτήματα σε οποιαδήποτε «διεύθυνση IP».
- ❖ Ενεργοποιεί μια «εξαίρεση» (firewall exception) στο τείχος προστασίας των λειτουργικών συστημάτων «Windows» που αφορά στις επικοινωνίες που λαμβάνουν χώρα μέσω του «WS-Management».
- ❖ Δημιουργεί τις απλές και πιο σύνθετες παραμετροποιήσεις του τελικού σημείου εάν χρειάζεται.
- ❖ Ενεργοποιεί όλες τις διαμορφώσεις της περιόδου σύνδεσης (session).
- ❖ Αλλάζει την περιγραφή ασφαλείας όλων των διαμορφώσεων περιόδου λειτουργίας (session), έτσι ώστε να επιτρέπεται η απομακρυσμένη πρόσβαση.
- ❖ Πραγματοποιεί επανεκκίνηση της υπηρεσίας «WinRM» για να κάνει τις προηγούμενες αλλαγές αποτελεσματικές και δεκτές από το σύστημα.

Αποτέλεσμα των παραπάνω φαίνεται στην εικόνα:

```
PS C:\WINDOWS\system32> Enable-PSRemoting
WinRM has been updated to receive requests.
WinRM service type changed successfully.
WinRM service started.

WinRM has been updated for remote management.
WinRM firewall exception enabled.

WARNING: Waiting for service 'Windows Remote Management (WS-Management) (winrm)' to stop...
```

Εικόνα 27: Αποτέλεσμα εκτέλεσης του «cmdlet Enable-PSRemoting».

Υπάρχουν δύο βασικά στοιχεία της υπηρεσίας «WinRM» που διέπουν τον τρόπο διασύνδεσης του «Ansible» με τον υπολογιστή με λειτουργικό σύστημα «Windows»: ο ακροατής (listener) και οι ρυθμίσεις διαμόρφωσης υπηρεσίας. Η υπηρεσία «WinRM» «ακούει» αιτήματα σε μία ή περισσότερες θύρες (ports). Κάθε μία από αυτές τις θύρες πρέπει να έχει έναν κατάλληλα

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

διαμορφωμένο «ακροατή». Προκειμένου να γίνει έλεγχος των «ακροατών» που εκτελούνται στην υπηρεσία «WinRM», εκτελείται η ακόλουθη εντολή στη διεπαφή «PowerShell» με δικαιώματα διαχειριστή:

```
winrm enumerate winrm/config/Listener
```

Το αποτέλεσμα που προκύπτει έχει ως εξής:

```
PS C:\Users\admin> winrm enumerate winrm/config/Listener
Listener
  Address = *
  Transport = HTTPS
  Port = 5986
  Hostname
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint = 100914AE523D95AC11F33F97D16EF9DA70315E19
  ListeningOn = 127.0.0.1, 169.254.247.102, 192.168.1.8, 192.168.162.3, ::1, 2a02:587:db1:3e82:6061:d579:9968:51c4, 2a02:587:db1:3e82:bcab:6a8e:e5de:8968, fe80::6061:d579:9968:51c4%10, fe80::91dd:61b:7fc2:f766%7

Listener [Source="Compatibility"]
  Address = *
  Transport = HTTP
  Port = 80
  Hostname
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint
  ListeningOn = 127.0.0.1, 169.254.247.102, 192.168.1.8, 192.168.162.3, ::1, 2a02:587:db1:3e82:6061:d579:9968:51c4, 2a02:587:db1:3e82:bcab:6a8e:e5de:8968, fe80::6061:d579:9968:51c4%10, fe80::91dd:61b:7fc2:f766%7

Listener [Source="Compatibility"]
  Address = *
  Transport = HTTPS
  Port = 443
  Hostname = WIN10PRO
  Enabled = true
  URLPrefix = wsman
  CertificateThumbprint
  ListeningOn = 127.0.0.1, 169.254.247.102, 192.168.1.8, 192.168.162.3, ::1, 2a02:587:db1:3e82:6061:d579:9968:51c4, 2a02:587:db1:3e82:bcab:6a8e:e5de:8968, fe80::6061:d579:9968:51c4%10, fe80::91dd:61b:7fc2:f766%7
```

Εικόνα 28: Ακροατές «WinRM (WinRM Listeners)

Στην Εικόνα 28 που εξήχθη από τη διεπαφή «PowerShell» της εικονικής μηχανής με λειτουργικό σύστημα «Windows 10» (Windows 10 host VM), παρατηρείται ότι είναι ενεργοποιημένοι δύο (2) «ακροατές» (listeners), όπου ο ένας «ακούει» στη θύρα «5985» μέσω του πρωτοκόλλου «HTTP» και ο άλλος «ακούει» στη θύρα «5986» μέσω του πρωτοκόλλου «HTTPS».

Βασικές έννοιες που αξίζουν να αναλυθούν, για την καλύτερη κατανόηση του παραπάνω αποτελέσματος είναι οι εξής:

Μεταφορά (Transport): Είτε ο «ακροατής» εκτελείται μέσω του πρωτοκόλλου «HTTP», είτε μέσω του πρωτοκόλλου «HTTPS», συνιστάται η χρήση ενός «ακροατή» μέσω του πρωτοκόλλου

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

«HTTPS», καθώς με τη χρήση του πρωτοκόλλου αυτού τα δεδομένα κρυπτογραφούνται χωρίς να απαιτούνται περαιτέρω αλλαγές.

Θύρα (Port): Η θύρα στην οποία εκτελείται ο «ακροατής», από προεπιλογή είναι η θύρα «5985» για το πρωτόκολλο «HTTP» και η θύρα «5986» για το πρωτόκολλο «HTTPS». Αυτή η θύρα μπορεί να αλλάξει παραμετροποιώντας τη μεταβλητή «ansible_port» στο μηχανήμα ελέγχου του «Ansible» (Ansible control machine). Στην παρούσα εργασία χρησιμοποιούνται οι προεπιλεγμένες θύρες.

URL Prefix: Το πρόθεμα «URL» για ακρόαση, από προεπιλογή είναι «wsman». Εάν αυτό αλλάξει στον υπολογιστή – στόχο, τότε θα πρέπει να ορισθεί και στη μεταβλητή «ansible_winrm_path» η ίδια τιμή.

CertificateThumbprint: Εάν ο «ακροατής» εκτελείται μέσω του πρωτοκόλλου «HTTPS», τότε δημιουργείται το «αποτύπωμα» του πιστοποιητικού στο «Windows Certificate Store», το οποίο χρησιμοποιείται στη σύνδεση και είναι μοναδικό για κάθε υπολογιστή – στόχο.

3.2.3.2 Δημιουργία - Παραμετροποίηση των ακροατών της υπηρεσίας «WinRM» (WinRM Listeners)

Υπάρχουν τρεις (3) τρόποι για να δημιουργηθούν και να παραμετροποιηθούν οι ακροατές της υπηρεσίας «WinRM»:

1ος Τρόπος: Χρήση «WinRM quickconfig» για το πρωτόκολλο «HTTP» ή «WinRM quickconfig – transport https» για το πρωτόκολλο «HTTPS».

Αυτός ο τρόπος είναι η ευκολότερη επιλογή σε περίπτωση που η υπηρεσία «WinRM» εκτελείται εκτός περιβάλλοντος δικτύου τομέα (domain network) και απαιτείται απλά η ύπαρξη ενός απλού ακροατή. Σε αντίθεση με τις άλλες επιλογές, αυτή η διαδικασία έχει το πρόσθετο πλεονέκτημα της δημιουργίας σχετικού κανόνα στο τείχος προστασίας για τις απαιτούμενες θύρες και εκκινεί την υπηρεσία «WinRM».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

2ος Τρόπος: Χρήση αντικειμένων πολιτικής ομάδας (Group Policy Objects).

Αυτός είναι ο καλύτερος τρόπος για να δημιουργηθεί ένας «ακροατής», όταν ο υπολογιστής αποτελεί υπολογιστή - πελάτη ενός δικτύου τομέα (domain network), κι αυτό διότι η διαμόρφωση πραγματοποιείται αυτόματα μέσα από τις δυνατότητες του «Windows Domain Controller Server». Στη συγκεκριμένη διπλωματική εργασία δεν εφαρμόστηκε αυτός ο τρόπος, διότι η εικονική μηχανή με λειτουργικό σύστημα «Windows 10», αρχικά, δεν αποτελεί μέλος κάποιου δικτύου τομέα (domain network).

Σε έναν οργανισμό η λύση αυτή θα ήταν η πλέον ενδεδειγμένη για την εγκατάσταση και παραμετροποίηση του συνόλου των απαιτήσεων της υπηρεσίας «WinRM» και των «ακροατών» της, προκειμένου όλα τα μέλη του δικτύου τομέα (domain network) να μπορούν να παραμετροποιηθούν αυτόματα και σχεδόν ταυτόχρονα, χωρίς την απαίτηση ανθρώπινης παρέμβασης σε κάθε Η/Υ πελάτη του εν λόγω δικτύου.

3ος Τρόπος: Χρήση της διεπαφής «PowerShell».

Στην εν λόγω διπλωματική εργασία, εφαρμόστηκε η μέθοδος αυτή. Οι εντολές που χρησιμοποιούνται για τη δημιουργία και την παραμετροποίηση ενός «ακροατή» μέσω του πρωτοκόλλου «HTTPS» για την υπηρεσία «WinRM», εκτελούνται στη διεπαφή «PowerShell» με **δικαιώματα διαχειριστή**, και έχουν ως εξής:

```
$selector_set = @{  
    Address = "*"   
    Transport = "HTTPS"  
}  
$value_set = @{  
    CertificateThumbprint = "100914AE523D95AC11F33F97D16EF9DA70315E19"  
}  
  
New-WSManInstance -ResourceURI "winrm/config/Listener" -SelectorSet $selector_set -  
ValueSet $value_set
```

CertificateThumbprint: Εκεί τοποθετήθηκε το «αποτύπωμα» που προέκυψε από την εκτέλεση της εντολής «winrm enumerate winrm/config/Listener», αφού όπως προαναφέρθηκε,

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

χρησιμοποιήθηκαν οι προεπιλεγμένες τιμές και δεν δημιουργήθηκαν εκ νέου πιστοποιητικά ασφαλείας στο «Windows Certificate Store» που χρησιμοποιείται στη σύνδεση από τον ακροατή της υπηρεσίας «WinRM» μέσω του πρωτοκόλλου «HTTPS»⁶⁷.

⁶⁷<https://docs.microsoft.com/el-gr/previous-versions/windows/desktop/Policy/group-policy-objects?redirectedfrom=MSDN>·
https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html#upgrading-powershell-and-net-framework

3.2.3.3 Παραμετροποιήσεις της υπηρεσίας «WinRM»

Υπάρχουν πολλές παράμετροι που μπορούν να ρυθμιστούν για τον έλεγχο της συμπεριφοράς της υπηρεσίας «WinRM», συμπεριλαμβανομένων των επιλογών ελέγχου ταυτότητας και των ρυθμίσεων μνήμης.

Προκειμένου να ληφθούν οι τρέχουσες επιλογές διαμόρφωσης υπηρεσίας, εκτελούνται στη διεαπφή «PowerShell» με δικαιώματα διαχειριστή, οι ακόλουθες εντολές:

```
winrm get winrm/config/Service
```

```
winrm get winrm/config/Winrs
```

Από την εκτέλεση των παραπάνω εντολών, προκύπτει:

```
PS C:\Users\admin> winrm get winrm/config/Service
Service
RootSDDL = O:NSG:BAD:P(A;;GA;;;BA)(A;;GR;;;IU)S:P(AU;FA;GA;;;WD)(AU;SA;GXGW;;;WD)
MaxConcurrentOperations = 4294967295
MaxConcurrentOperationsPerUser = 1500
EnumerationTimeoutms = 240000
MaxConnections = 300
MaxPacketRetrievalTimeSeconds = 120
AllowUnencrypted = false
Auth
  Basic = true
  Kerberos = true
  Negotiate = true
  Certificate = false
  CredSSP = false
  CbtHardeningLevel = Relaxed
DefaultPorts
  HTTP = 5985
  HTTPS = 5986
IPv4Filter = *
IPv6Filter = *
EnableCompatibilityHttpListener = true
EnableCompatibilityHttpsListener = true
CertificateThumbprint
AllowRemoteAccess = true

PS C:\Users\admin> winrm get winrm/config/Winrs
Winrs
AllowRemoteShellAccess = true
IdleTimeout = 7200000
MaxConcurrentUsers = 2147483647
MaxShellRunTime = 2147483647
MaxProcessesPerShell = 2147483647
MaxMemoryPerShellMB = 2147483647
MaxShellsPerUser = 2147483647
```

Εικόνα 29: Ρυθμίσεις υπηρεσίας «WinRM».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Κάποιες από τις σημαντικότερες παραμετροποιήσεις της υπηρεσίας «WinRM» είναι οι παρακάτω⁶⁸:

Service\AllowUnencrypted: Η επιλογή αυτή καθορίζει εάν η υπηρεσία «WinRM» θα επιτρέπει την κυκλοφορία που εκτελείται μέσω του πρωτοκόλλου «HTTP» χωρίς κρυπτογράφηση μηνυμάτων. Η κρυπτογράφηση επιπέδου μηνύματος είναι δυνατή μόνο όταν η μεταβλητή «ansible_winrm_transport» λαμβάνει τιμές «ntlm», «kerberos» ή «credssp», οι οποίες αποτελούν πρωτόκολλα αυθεντικοποίησης σε λειτουργικά συστήματα «Windows». Από προεπιλογή, αυτό είναι «ψευδές» (false) και θα πρέπει να οριστεί ως «αληθές» (true) μόνο κατά τον εντοπισμό σφαλμάτων μηνυμάτων (debugging) της υπηρεσίας «WinRM». Όπως φαίνεται και στην *Εικόνα 29* χρησιμοποιήθηκε στην παρούσα μελέτη η τιμή «false».

Service\ Auth*: Αυτές οι σημάνσεις (flags) καθορίζουν ποιες επιλογές ελέγχου ταυτότητας (authentication) επιτρέπονται με την υπηρεσία «WinRM». Από προεπιλογή, τα πρωτόκολλα «Negotiate (NTLM)» και «Kerberos» είναι ενεργοποιημένα. Το ίδιο ισχύει και στη συγκεκριμένη μελέτη.

Service\Auth\CbtHardeningLevel: Καθορίζει εάν τα διακριτικά δέσμευσης καναλιού (Channel Binding Tokens - cbt) δεν επαληθεύονται (Κανένα - None), επαληθεύονται αλλά δεν απαιτείται επαλήθευση (Χαλαρωμένο - Relaxed) ή επαληθεύονται και απαιτείται η επαλήθευσή τους (Αυστηρό - Strict). Τα «CBT» χρησιμοποιούνται μόνο όταν η σύνδεση πραγματοποιείται μέσω των πρωτοκόλλων «NTLM» ή «Kerberos» μέσω «HTTPS». Στην εν λόγω διπλωματική εργασία, η τιμή έχει ορισθεί ως «Relaxed».

Service\CertificateThumbprint: Αυτό αποτελεί το αποτύπωμα του πιστοποιητικού που χρησιμοποιείται για την κρυπτογράφηση του καναλιού «TLS», που χρησιμοποιείται με το πρωτόκολλο ελέγχου ταυτότητας «CredSSP». Από προεπιλογή, αυτό είναι κενό. Ένα αυτο-

⁶⁸https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html#upgrading-powershell-and-net-framework

υπογεγραμμένο (self-signed) πιστοποιητικό δημιουργείται όταν εκκινεί η υπηρεσία «WinRM» και χρησιμοποιείται στη διαδικασία «TLS» (TLS process).

Winrs\MaxShellRunTime: Αυτός είναι ο μέγιστος χρόνος, σε χιλιοστά του δευτερολέπτου (ms), που επιτρέπεται να εκτελείται μια απομακρυσμένη εντολή.

Winrs\MaxMemoryPerShellMB: Αυτή είναι η μέγιστη ποσότητα μνήμης που διατίθεται ανά κέλυφος (shell), συμπεριλαμβανομένων και των θυγατρικών διεργασιών του κελύφους.

3.2.3.4 Συνήθη προβλήματα της υπηρεσίας «WinRM» - Αντιμετώπιση

Καθόσον η υπηρεσία «WinRM» διαθέτει ένα ευρύ φάσμα επιλογών διαμόρφωσης, μπορεί να είναι δύσκολο να ρυθμιστεί και να παραμετροποιηθεί. Λόγω αυτής της πολυπλοκότητας, τα ζητήματα που προκύπτουν κατά τη χρήση του «Ansible» όταν χρησιμοποιείται για τη διαμόρφωση λειτουργικών συστημάτων «Windows», θα μπορούσαν στην πραγματικότητα να οφείλονται σε πιθανώς λανθασμένη παραμετροποίηση της υπηρεσίας «WinRM».

Ένας εύκολος τρόπος για να προσδιορισθεί εάν ένα πρόβλημα οφείλεται στην υπηρεσία «WinRM», είναι η εκτέλεση εντολών ελέγχου, που αναφέρονται παρακάτω, από έναν υπολογιστή με λειτουργικό σύστημα «Windows» προς έναν άλλο με ίδιο λειτουργικό, χωρίς την παρεμβολή του μηχανήματος ελέγχου του «Ansible».

```
# Έλεγχος του HTTP
winrs -r:http://server:5985/wsman -u:Username -p:Password ipconfig

# Έλεγχος του HTTPS
winrs -r:https://server:5986/wsman -u:Username -p:Password -ssl ipconfig

# Έλεγχος του HTTPS, αγνοώντας την επαλήθευση πιστοποιητικού
$username = "Username"
$password = ConvertTo-SecureString -String "Password" -AsPlainText -Force
$cred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $username, $password

$session_option = New-PSSessionOption -SkipCACheck -SkipCNCheck -SkipRevocationCheck
```

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

<code>Invoke-Command -ComputerName server -UseSSL -ScriptBlock { ipconfig } -Credential \$cred -SessionOption \$session_option</code>

Εάν αυτή η μέθοδος αποτύχει τότε το ζήτημα που έχει προκύψει, πιθανώς να σχετίζεται με την παραμετροποίηση της υπηρεσίας «WinRM». Κάποια από τα συνήθη προβλήματα που σχετίζονται με την υπηρεσία «WinRM», είναι τα εξής:

HTTP 401 / Credentials Rejected

Ο κωδικός σφάλματος «HTTP 401» υποδεικνύει ότι η διαδικασία ελέγχου ταυτότητας απέτυχε κατά την αρχική σύνδεση. Στην περίπτωση αυτή θα πρέπει να διενεργηθούν οι παρακάτω έλεγχοι:

- Έλεγχος ότι τα διαπιστευτήρια είναι σωστά και ορθά ορισμένα στο αρχείο «inventory» με τη χρήση των μεταβλητών «ansible_user» και «ansible_password».
- Έλεγχος ότι ο χρήστης που ορίστηκε στο αρχείο «inventory» είναι μέλος της τοπικής ομάδας διαχειριστών (local administrators group) ή του έχει δοθεί ρητή πρόσβαση (μπορεί να χρησιμοποιηθεί ένας έλεγχος σύνδεσης με την εντολή «winrs» για να αποκλειστεί αυτό).
- Έλεγχος εάν η επιλογή ελέγχου ταυτότητας που ορίστηκε στη μεταβλητή «ansible_winrm_transport» είναι ενεργοποιημένη και στο «Service\ Auth*». Εάν η υπηρεσία «WinRM» εκτελείται μέσω πρωτοκόλλου «HTTP» και όχι «HTTPS», πρέπει να χρησιμοποιούνται τα πρωτόκολλα αυθεντικοποίησης «ntlm», «Kerberos» ή «credssp» με την τιμή «auto» στη μεταβλητή «ansible_winrm_message_encryption», για να ενεργοποιηθεί η κρυπτογράφηση μηνυμάτων. Εάν χρησιμοποιείται άλλη επιλογή ελέγχου ταυτότητας ή εάν δεν είναι δυνατή η αναβάθμιση της εγκατεστημένης έκδοσης της βιβλιοθήκης «pywinrm» στο μηχάνημα ελέγχου του «Ansible», το «Service\AllowUnencrypted» μπορεί να οριστεί σε «true», αλλά αυτό συνιστάται μόνο για την αντιμετώπιση προβλημάτων.

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

- Εάν χρησιμοποιείται για τον έλεγχο ταυτότητας το πρωτόκολλο αυθεντικοποίησης «Kerberos», πρέπει να γίνει έλεγχος εάν το «Service\Auth\CbtHardeningLevel» έχει οριστεί σε «Strict». Εάν ναι, θα πρέπει να ορισθεί σε «Relaxed».
- Όταν χρησιμοποιείται για τον έλεγχο ταυτότητας το πρωτόκολλο «Basic» ή κάποιο πιστοποιητικό, πρέπει να γίνει έλεγχος εάν ο χρήστης αποτελεί τοπικό λογαριασμό (local account) και όχι λογαριασμό δικτύου τομέα (domain account). Οι λογαριασμοί τομέα δεν λειτουργούν με τον έλεγχο ταυτότητας μέσω του πρωτοκόλλου «Basic» ή με τη χρήση πιστοποιητικού (certificate authentication).

Σφάλμα HTTP 500

Το σφάλμα αυτό σχετίζεται με την υπηρεσία «WinRM» και θα πρέπει να ελεγχθεί εάν ο αριθμός των τρέχοντων ανοικτών κελυφών δεν έχει ξεπεράσει την τιμή που ορίζεται στο «WinRsMaxShellsPerUser» ή κανένα από τα άλλα όρια που ορίζονται στο «Winrs».

Σφάλματα χρονικού ορίου (Timeout errors)

Με την εμφάνιση των εν λόγω σφαλμάτων υποδηλώνονται ζητήματα αναφορικά με τη σύνδεση δικτύου όπου το μηχάνημα ελέγχου του «Ansible» (Ansible control machine) δεν μπορεί να επικοινωνήσει με τον υπολογιστή - στόχο. Στην περίπτωση αυτή απαιτείται έλεγχος των κάτωθι:

- Έλεγχος εάν το τείχος προστασίας δεν έχει ρυθμιστεί για να αποκλείονται οι διαμορφωμένες θύρες του «ακροατή WinRM» (WinRM Listener).
- Έλεγχος εάν η θύρα του «ακροατή WinRM» που έχει ορισθεί στο λειτουργικό σύστημα «Windows», ταυτίζεται με αυτή που έχει ορισθεί στο μηχάνημα ελέγχου του «Ansible» (Ansible control machine).
- Έλεγχος εάν η υπηρεσία «WinRM» έχει ορισθεί να εκτελείται αυτόματα κατά την εκκίνηση των λειτουργικών συστημάτων «Windows».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Σφάλματα απόρριψης της σύνδεσης (Connection Refused Errors)

Υποδηλώνουν σφάλματα κατά την προσπάθεια επικοινωνίας μέσω της υπηρεσίας «WinRM».

Θα πρέπει να διεξαχθούν οι εξής έλεγχοι :

- Έλεγχος ότι η υπηρεσία «WinRM» λειτουργεί στον υπολογιστή - στόχο. Αυτό μπορεί να υλοποιηθεί με τη χρήση της εντολής «Get-Service -Name winrm», η οποία θα φέρει ως αποτέλεσμα την κατάσταση της υπηρεσίας «WinRM» στο συγκεκριμένο υπολογιστή τη δεδομένη χρονική στιγμή.
- Επιπλέον, πρέπει να ελεγχθεί ότι το τείχος προστασίας του υπολογιστή επιτρέπει την κυκλοφορία μέσω της θύρας που έχει ορισθεί για τον «ακροατή WinRM» (WinRM Listener). Από προεπιλογή, αυτές οι θύρες είναι η «5985» για το πρωτόκολλο «HTTP» και η «5986» για το πρωτόκολλο «HTTPS».
- Τέλος, σημειώνεται ότι ορισμένες φορές ένα πρόγραμμα εγκατάστασης λογισμικού ενδέχεται να επανεκκινήσει την υπηρεσία «WinRM» και να προκαλέσει αυτό το σφάλμα. Γενικά, η υπηρεσία «WinRM» είναι αρκετά «ευαίσθητη» σε οποιαδήποτε μεταβολή που έχει να κάνει με τη σύνδεση δικτύου. Ένας τρόπος αντιμετώπισης του συγκεκριμένου προβλήματος είναι η αλλαγή των προεπιλεγμένων τιμών των μεταβλητών «ansible_wirm_operation_timeout_sec» και «ansible_wirm_read_timeout_sec» σε μεγαλύτερες τιμές. Οι προεπιλεγμένες τιμές των μεταβλητών αυτών είναι 20 s για την πρώτη και 60 s για τη δεύτερη.

3.3 Επικοινωνία εργαλείου «Ansible» και εικονικών μηχανών «Windows»

Ύστερα από την τεχνική ανάπτυξη της υποδομής για τις ανάγκες εκπόνησης της παρούσας διπλωματικής εργασίας, η οποία και αναλύθηκε εκτενώς στο Κεφάλαιο 3, κρίνεται αναγκαία η δοκιμή της επικοινωνίας μεταξύ του μηχανήματος ελέγχου του «Ansible» (Ansible Control

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Machine) και των εικονικών μηχανών που διαθέτουν λειτουργικά συστήματα εκδόσεων «Windows».

Επομένως, η δοκιμή αυτή υλοποιήθηκε μέσω της εντολής:

```
ansible winhost -i inventory.ini -m win_ping --ask-vault-pass
```

Όπου:

winhost: Είναι ο εικονικός υπολογιστής – στόχος όπως αναλύεται και στο αρχείο «inventory» ([Παράγραφος 3.2.2.2](#)).

Η παράμετρος -i: Με τη χρήση της παραμέτρου αυτής ορίζεται το αρχείο «inventory» που χρησιμοποιείται. Στην προκειμένη περίπτωση είναι το αρχείο «inventory.ini».

Η παράμετρος -m: Με την παράμετρο αυτή, ορίζεται η χρήση ενός από τα «modules» που διαθέτει το «Ansible» για την αυτοματοποιημένη παραμετροποίηση ενός υπολογιστή με λειτουργικό σύστημα «Windows».

To module «win_ping»: Χρησιμοποιείται προκειμένου να ελεγχθεί η δικτυακή επικοινωνία μεταξύ του μηχανήματος ελέγχου του «Ansible» (Ansible Control Machine) και του υπολογιστή - στόχου με λειτουργικό σύστημα «Windows 10». Με το «module» αυτό ουσιαστικά εκτελείται στον υπολογιστή – στόχο μέσω της διεπαφής «PowerShell» η εντολή «ping <ip address>».

Η παράμετρος «--ask-vault-pass»: Η χρήση της παραμέτρου αυτής είναι απαραίτητη, καθώς όπως προαναφέρθηκε στην [παράγραφο 3.2.2.3](#), έχει πραγματοποιηθεί κρυπτογράφηση με τη χρήση του «Ansible Vault». Συνεπώς, για να εκτελεστεί η εντολή ορθά, θα πρέπει να χρησιμοποιηθεί η συγκεκριμένη παράμετρος.

Το επιτυχές αποτέλεσμα έχει ως εξής:

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

```

debuseradmin@debian:~$ ansible winhost -i inventory.ini -m win_ping --ask-vault-pass
Vault password:
192.168.162.3 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
debuseradmin@debian:~$ █

```

Εικόνα 30: Επιτυχής παραμετροποίηση και επικοινωνία μεταξύ του εργαλείου «Ansible» και των λειτουργικών συστημάτων «Windows».

Με την εξασφάλιση της επιτυχούς επικοινωνίας, είναι δυνατή η περαιτέρω δοκιμή και μελέτη διαφόρων περιπτώσεων αυτοματοποιημένης παραμετροποίησης ενός υπολογιστή, ο οποίος διαθέτει λειτουργικό σύστημα των εκδόσεων και διανομών των «Microsoft Windows», με τη χρήση του «Ansible», όπως θα γίνει και στη συνέχεια.

Κεφάλαιο 4^ο: Μελέτες Περίπτωσης (Case Studies)

4.1 Εγκατάσταση λογισμικού «Android Studio» και εργαλείων «SDK» (SDK Tools)

Μια εκ των μελετών περίπτωσης αφορά στην εγκατάσταση του πακέτου λογισμικού «Android Studio» και των εργαλείων «SDK» (SDK Tools). Το λογισμικό «Android Studio» είναι το επίσημο «IDE» λογισμικό που χρησιμοποιείται για την ανάπτυξη εφαρμογών «Android». Παρέχει τα βασικά εργαλεία που απαιτούνται για να ξεκινήσει η ανάπτυξη εφαρμογών «Android», όπως το «Android Studio IDE» και τα εργαλεία «SDK Android – SDK Tools». Το πακέτο των εργαλείων «SDK» (SDK Tools) έχει καταργηθεί και δεν λαμβάνει πλέον ενημερώσεις. Αντ' αυτού, χρησιμοποιείται το νέο πακέτο εργαλείων γραμμής εντολών (command-line tools)⁶⁹.

Επιπλέον, παύει ο παλιός τρόπος ενημέρωσης αντικειμένων του εργαλείου «SDK Manager». Αυτή η αλλαγή επηρεάζει μόνο την έκδοση «Android Studio 2.2» και τις χαμηλότερες εκδόσεις. Στο μέλλον, δεν θα δημοσιεύονται πλέον ενημερώσεις με την παλαιότερη μορφή «XML». Οι χρήστες παλαιότερων εκδόσεων του λογισμικού «Android Studio», της παλιάς γραμμής εντολών του διαχειριστή «SDK» (SDK Manager) ή της παλιάς διεπαφής χρήστη «SDK Manager UI» δεν θα λαμβάνουν ενημερώσεις για τα εργαλεία «SDK» (SDK Tools) μέσω του «SDK Manager». Παρόλο

⁶⁹ <https://developer.android.com/studio/releases/sdk-tools>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

που οι τρέχουσες εκδόσεις θα συνεχίσουν να λειτουργούν, τα εργαλεία «SDK» (SDK Tools) θα ενημερώνονται με τη χρήση των νέων εργαλείων γραμμής εντολών (command-line tools)⁷⁰.

Εάν εγκατασταθεί το λογισμικό «Android Studio 2.2» ή νεότερη έκδοση του, όπως έγινε και στην συγκεκριμένη μελέτη περίπτωσης, δεν είναι πλέον εφικτή η εκ νέου λήψη των εργαλείων «SDK» (SDK Tools). Για τη λήψη των στοιχείων μέσω του διαχειριστή «SDK» (SDK Manager), θα πρέπει να πραγματοποιηθεί αναβάθμιση σε νεότερη έκδοση του λογισμικού «Android Studio»⁷¹.

Σημειώνεται ότι, πλέον, μέσω της εκτέλεσης του εγκατεστημένου λογισμικού «Android Studio», ο εκάστοτε χρήστης, μπορεί να εγκαταστήσει τα εργαλεία «SDK» (SDK tools). Στη συγκεκριμένη περίπτωση όμως, η εγκατάσταση αυτών πραγματοποιήθηκε αυτοματοποιημένα με τη χρήση του εργαλείου «Ansible». Με τον τρόπο αυτό, ο τελικός χρήστης (end user) τα βρίσκει «έτοιμα προς χρήση», μαζί με το λογισμικό «Android Studio», μόλις κάνει έναρξη του προγράμματος. Για να επιτευχθεί αυτό, χρειάστηκε μέσω του «playbook», να εκτελεστούν κάποιες διαδικασίες στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10».

Κρίνεται αναγκαίο να αναφερθεί ότι για την ορθή λειτουργία τόσο του λογισμικού «Android Studio», όσο και των εργαλείων «SDK» (SDK Tools), απαραίτητη προϋπόθεση αποτελεί η εγκατάσταση της γλώσσας «Java», καθώς και ο ορισμός των μεταβλητών περιβάλλοντος που απαιτούνται για τη λειτουργία της.⁷² Για το σκοπό αυτό, χρησιμοποιήθηκε το «Open jdk 14.0.1».

Προκειμένου να υλοποιηθεί η εν λόγω διαδικασία με τη χρήση του εργαλείου «Ansible» στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10», δημιουργήθηκε ένα αρχείο «playbook» με το οποίο πραγματοποιούνται όλες οι σχετικές απαραίτητες διαδικασίες. Οι διαδικασίες αυτές έχουν ως εξής:

1. Αρχικά, δημιουργήθηκε το αρχείο «AndroidStudio.yml» σε ένα φάκελο που ονομάστηκε «Playbooks», στη διαδρομή «/home/debuseradmin/Playbooks». Στο «playbook» αυτό τοποθετήθηκαν με διαδοχική σειρά εκτέλεσης, όλα τα «modules» του «Ansible», καθώς

⁷⁰ <https://developer.android.com/studio/intro/update#sdk-manager>

⁷¹ <https://developer.android.com/studio>, <https://developer.android.com/studio/releases/sdk-tools>

⁷² https://www.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.html

επίσης και οποιεσδήποτε επιπλέον παραμετροποιήσεις, εντολές και αλλαγές που απαιτήθηκαν για την υλοποίηση της εγκατάστασης του λογισμικού «Android Studio» και των εργαλείων «SDK» (SDK Tools) στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10».

Η δημιουργία του αρχείου «AndroidStudio.yml», πραγματοποιήθηκε μέσα στον προαναφερόμενο φάκελο με τη χρήση των εξής εντολών διαδοχικά, στο τερματικό (Terminal) χρησιμοποιώντας τον κειμενογράφο «Vim»:

```
cd /home/debuseradmin/playbooks  
vim AndroidStudio.yml
```

Η εκτέλεση έχει ως εξής, όπως φαίνεται και στην εικόνα παρακάτω:



```
debuseradmin@debian:~$ cd /home/debuseradmin/playbooks/  
debuseradmin@debian:~/playbooks$ vim AndroidStudio.yml
```

Εικόνα 31: Δημιουργία αρχείου «Playbook» για την εγκατάσταση του «Android Studio».

2. Στη συνέχεια, έγινε η συγγραφή του αρχείου «playbook», εισάγοντας αρχικά:

```
---  
  
- name: install Android Studio  
  hosts: winhost
```

Όπου:

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

- **Οι τρεις (3) παύλες στην αρχή:** Υποδηλώνουν αρχείο το οποίο είναι τύπου «.yml», και συντάσσεται με τη δηλωτική γλώσσα «YAML»⁷³, την οποία χρησιμοποιεί το «Ansible», όπως προαναφέρθηκε.
- **hosts:** Μεταβλητή της οποίας η τιμή υποδηλώνει το όνομα του υπολογιστή – στόχου (host), όπως αυτός έχει δηλωθεί στο αρχείο «inventory» που περιγράφεται στην παράγραφο 3.2.2.2.

3. Στη συνέχεια, πραγματοποιήθηκε εγκατάσταση του λογισμικού «Android Studio», το οποίο για να εγκατασταθεί στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10», χρησιμοποιείται το «url» του «.exe» αρχείου, από την επίσημη ιστοσελίδα που παρέχει το λογισμικό «Android Studio» και τα λοιπά εργαλεία που χρησιμοποιούνται με αυτό⁷⁴. Η εγκατάσταση του λογισμικού «Android Studio» πραγματοποιήθηκε με τη χρήση του «module» «win_package»⁷⁵ του εργαλείου «Ansible», όπως φαίνεται και παρακάτω, στο πρώτο μέρος του «playbook»:

```
tasks:
- name: Install AS
  win_package:
    path: https://redirector.gvt1.com/edgedl/android/studio/install/3.6.2.0/android-studio-ide-192.6308749-windows.exe
    product_id: android-studio
    arguments: /NCRC /S /IACCEPTTEULA
  ignore_errors: true
  failed_when: false
  changed_when: false
```

⁷³ https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html

⁷⁴ <https://redirector.gvt1.com/edgedl/android/studio/install/3.6.2.0/android-studio-ide-192.6308749-windows.exe>

⁷⁵ https://docs.ansible.com/ansible/latest/modules/win_package_module.html

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Όπου:

- **win_package**: Αποτελεί ενότητα λογισμικού (module), το οποίο χρησιμοποιείται για την εγκατάσταση ή την απεγκατάσταση πακέτων λογισμικού των τύπων «.exe» και «.msi» σε υπολογιστές – στόχους (hosts) με λειτουργικό σύστημα «Windows».
- **path**: Η μεταβλητή αυτή, δηλώνει τη θέση του πακέτου που πρόκειται να εγκατασταθεί ή να απεγκατασταθεί. Τα πακέτα λογισμικού μπορεί να βρίσκονται είτε σε μια δικτυακή τοποθεσία (network file share), είτε σε κάποιον τοπικό φάκελο, είτε να ληφθούν από το διαδίκτυο. Εάν η διαδρομή βρίσκεται σε μια δικτυακή τοποθεσία (network file share), και η τρέχουσα μεταφορά της υπηρεσίας «WinRM» (μεταβλητή winrm_transport στο αρχείο «inventory») δεν υποστηρίζει εκχώρηση διαπιστευτηρίων, τότε το όνομα χρήστη (μεταβλητή ansible_user στο αρχείο «inventory») και ο κωδικός πρόσβασης (μεταβλητή ansible_passwords στο αρχείο «inventory») πρέπει να ρυθμιστούν εκ νέου για πρόσβαση στο αρχείο.
- **product_id**: Το αναγνωριστικό προϊόντος του προς εγκατάσταση πακέτου λογισμικού. Αυτό χρησιμοποιείται για τον έλεγχο του εάν το προϊόν είναι ήδη εγκατεστημένο και για τη λήψη των πληροφοριών απεγκατάστασης.
- **arguments**: Οποιαδήποτε παράμετρος που απαιτείται από το πακέτο λογισμικού για να την πραγματοποιήσει εγκατάσταση χωρίς παρακολούθηση (unattended installation). Οι παράμετροι αυτοί έχουν ως εξής:
 - ❖ «/S», η οποία εκτελεί εγκατάσταση χωρίς επίβλεψη και αλληλεπίδραση (unattended installation - silent installation)
 - ❖ «/NCRC», η οποία χρησιμοποιείται για να μην εκτελεστεί ο έλεγχος «CRC», σε ό,τι αφορά «NSIS» εκτελέσιμα αρχεία, όπως είναι και το «Android Studio»⁷⁶.

⁷⁶<https://nsis.sourceforge.io/Docs/Chapter3.html>

❖ «/IACCEPTEULA», η οποία αποδέχεται άδειες χρήσης.

- **ignore errors**: Σημειώνεται ότι, παρόλο που η εγκατάσταση πραγματοποιήθηκε κανονικά και ο χρήστης που χρησιμοποιεί το «Ansible» είναι χρήστης με δικαιώματα διαχειριστή, κατά την ολοκλήρωση αυτής, εμφανιζόταν ο κωδικός σφάλματος «1223». Έπειτα από αναζήτηση του προβλήματος στο διαδίκτυο, διαπιστώθηκε ότι αυτό συμβαίνει σε περιπτώσεις όπου γίνεται αυτοματοποιημένη εγκατάσταση, και σχετίζεται με τα δικαιώματα του χρήστη που έχει οριστεί να εκτελέσει την εγκατάσταση⁷⁷. Για να επιτευχθεί αυτό, δοκιμάστηκε να οριστεί στο «playbook» η μέθοδος «become» με τιμή «runas» που ορίζει ότι οι εντολές που αναφέρονται στο «task» αυτό, θα εκτελεστούν σαν διαχειριστής⁷⁸. Αυτό δεν είχε αποτέλεσμα, και κατόπιν νέας αναζήτησης, προτιμήθηκε να αγνοηθεί το συγκεκριμένο σφάλμα με την χρήση της μεταβλητής «**ignore_errors**»⁷⁹ και ορίζοντας την τιμή της σε «αληθής» (true), όπως προαναφέρθηκε, αφού ουσιαστικά δεν επιφέρει καμία επίπτωση κατά την εγκατάσταση⁸⁰.
- **failed when**: Αποτελεί υπόθεση (conditional) με την οποία ορίζεται στο «Ansible» υπό ποιους όρους ορίζεται η αποτυχία. Ορίστηκε σε «ψευδής» (false) καθότι το παραπάνω σφάλμα θα αγνοηθεί όπως περιγράφηκε πιο πάνω. Με την τιμή «false» ορίζεται ότι το συγκεκριμένο «task» δεν θα αποτύχει και η εκτέλεση του των υπολοίπων διαδοχικών εργασιών (tasks) στο «Playbook» θα εκτελεστούν μετά την εν λόγω εργασία (task)⁸¹.
- **changed when**: Αποτελεί, επίσης, υπόθεση με την οποία ορίζεται εάν στην ολοκλήρωση της εκτέλεσης του «task» θα αναφερθεί κάποια αλλαγή. Ορίστηκε η τιμή «ψευδής» (false) προκειμένου να μην αναφέρεται ποτέ αλλαγή της

<https://superuser.com/questions/1432990/android-studio-nsis-error-while-installing>
https://nsis.sourceforge.io/Main_Page

⁷⁷ <https://stackoverflow.com/questions/46038621/how-to-install-android-sdk-to-all-users-silently-on-windows>

⁷⁸ https://docs.ansible.com/ansible/latest/user_guide/become.html#become-and-windows

⁷⁹ https://docs.ansible.com/ansible/latest/user_guide/playbooks_error_handling.html

⁸⁰ <https://github.com/PSAppDeployToolkit/PSAppDeployToolkit/issues/365>

⁸¹ https://docs.ansible.com/ansible/latest/user_guide/playbooks_error_handling.html#handlers-and-failure

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

κατάστασης του «task», εφόσον το «task» ορίστηκε να εκτελεστεί κανονικά και να αγνοηθεί το σφάλμα, όπως περιγράφηκε και παραπάνω.

4. Έπειτα, δημιουργήθηκε στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10» ένας φάκελος όπου θα αποθηκευτούν τα αρχεία που θα χρησιμοποιηθούν. Ο φάκελος αυτός δημιουργήθηκε με τη σειρά του στον σκληρό δίσκο, στο διαμέρισμα (partition) με όνομα «C:/» και ονομάστηκε «Programms». Αυτό έγινε με την εισαγωγή στο «playbook», αμέσως μετά το «task» για την εγκατάσταση του «Android Studio», της παρακάτω εργασίας (task) όπου για την υλοποίησή της χρησιμοποιήθηκε το «module» «win_file»⁸², όπως παρακάτω:

```
- name: Create Directory for download tools
  win_file:
    path: C:/Programms
    state: directory
```

Όπου:

- **win file:** Αποτελεί ενότητα λογισμικού (module) που χρησιμοποιείται για τη δημιουργία κενών αρχείων, για την ενημέρωση τροποποίησης ήδη υπαρχόντων αρχείων και για την δημιουργία και τη διαγραφή δομών φακέλων αρχείων (directories).
- **path:** Μεταβλητή στην οποία ορίζεται η διαδρομή στο προς διαχείριση αρχείο ή φάκελο.
- **state:** Μεταβλητή στην οποία ορίζεται η επιθυμητή λειτουργία. Στη εν λόγω περίπτωση, ορίστηκε η τιμή «directory». Η τιμή αυτή αναφέρεται στη δημιουργία μια δομής φακέλων στη διαδρομή που ορίστηκε στη μεταβλητή «path», στην οποία όλοι οι υποφάκελοι θα δημιουργηθούν εάν δεν υπάρχουν.

⁸² https://docs.ansible.com/ansible/latest/modules/win_file_module.html

5. Στη συνέχεια πραγματοποιήθηκε λήψη του συμπιεσμένου αρχείου που περιέχει τα επιθυμητά αρχεία από τον ιστότοπο της γλώσσας «Java»⁸³ και η αποθήκευσή του στον προαναφερόμενο φάκελο. Αυτό έγινε με την εισαγωγή της παρακάτω εργασίας (task) στο «playbook», ακριβώς αμέσως μετά το προηγούμενο «task» που αφορούσε στη δημιουργία φακέλου. Για την υλοποίηση της εν λόγω εργασίας χρησιμοποιήθηκε το «module» «win_get_url»⁸⁴ όπως περιγράφεται στη συνέχεια:

```
- name: Download Java
  win_get_url:
    url:
      https://download.java.net/java/GA/jdk14.0.1/664493ef4a6946b186ff29eb326336a2/7/GPL/openjdk-14.0.1_windows-x64_bin.zip
    dest: C:/Programms
```

Όπου:

- **win_get_url**: Αποτελεί ενότητα λογισμικού του «Ansible» (module), με τη χρήση του οποίου πραγματοποιούνται λήψεις αρχείων από ιστοτόπους όπου χρησιμοποιούνται πρωτόκολλα «HTTP» ή «HTTPS» ή από εξυπηρετητή «FTP» (FTP Server).
- **url**: Μεταβλητή η οποία εκφράζει την πλήρης διεύθυνση «URL» ενός αρχείου για λήψη.
- **dest**: Η τοποθεσία στην οποία θα αποθηκευτεί το αρχείο που θα ληφθεί.

6. Έπειτα, πραγματοποιήθηκε εξαγωγή των αρχείων από τη διαδρομή «C:/Programms/openjdk-14.0.1_windows-x64_bin.zip» και δημιουργία φακέλου στη

⁸³https://download.java.net/java/GA/jdk14.0.1/664493ef4a6946b186ff29eb326336a2/7/GPL/openjdk-14.0.1_windows-x64_bin.zip

⁸⁴https://docs.ansible.com/ansible/latest/modules/win_get_url_module.html

διαδρομή «C:\Program Files\Java», έτσι ώστε πλέον να υφίσταται η διαδρομή «C:\Program Files\Java\jdk-14.0.1», όπου και υπάρχουν όλα τα απαραίτητα αρχεία για τη χρήση της γλώσσας «Java». Αυτό πραγματοποιήθηκε με τη δημιουργία του αμέσως επόμενου «task», το οποίο πραγματοποιεί την εξαγωγή και δημιουργεί και τον φάκελο στην επιθυμητή διαδρομή προκειμένου η «Java» να μπορεί να χρησιμοποιηθεί από τα λειτουργικά συστήματα «Windows». Το συγκεκριμένο αρχείο που χρησιμοποιήθηκε για την εγκατάσταση της γλώσσας «Java», δεν απαιτεί κάποια επιπλέον εκτέλεση κάποιου πακέτου λογισμικού. Για την υλοποίηση της εξαγωγής του πακέτου γλώσσας «Java» στην επιθυμητή διαδρομή, χρησιμοποιήθηκε η ενότητα λογισμικού «win_unzip»⁸⁵, όπως περιγράφεται στη συνέχεια:

```
- name: Extract Java to correct place
win_unzip:
  src: C:/Programms/openjdk-14.0.1_windows-x64_bin.zip
  dest: C:/Program Files/Java
  creates: C:/Program Files/Java
```

⁸⁵ https://docs.ansible.com/ansible/latest/modules/win_unzip_module.html

Όπου:

- **win_unzip:** Ενότητα λογισμικού (module) που χρησιμοποιείται για την αποσυμπίεση αρχείων ή/και φακέλων σε κόμβους με λειτουργικό σύστημα «Windows».
- **src:** Μεταβλητή η οποία εκφράζει την ακριβή διαδρομή όπου βρίσκεται το προς αποσυμπίεση αρχείο.
- **dest:** Μεταβλητή με την οποία ορίζεται η τοποθεσία αποσυμπίεσης του προαναφερόμενου αρχείου.
- **creates:** Εάν η τοποθεσία αποσυμπίεσης δεν υπάρχει, το αρχείο δεν θα αποσυμπιεστεί. Οπότε προκειμένου να γίνει η αποσυμπίεση του αρχείου, χρησιμοποιείται η μεταβλητή «creates» στην οποία δίνεται ως τιμή, η τιμή της μεταβλητής «dest», διότι στην προκειμένη περίπτωση, η διαδρομή αυτή δεν υπήρχε από πριν.

7. Στη συνέχεια, έγινε λήψη των εργαλείων «SDK» (SDK Tools)⁸⁶ στη διαδρομή «C:/Programms». Αυτό πραγματοποιήθηκε με την ανάπτυξη της παρακάτω εργασίας (task) χρησιμοποιώντας την ενότητα λογισμικού (module) «win_get_url», η οποία περιεγράφηκε αναλυτικά στο βήμα 5, και στο συγκεκριμένο «task» αναπτύσσεται όπως παρακάτω:

```
- name: Download Command Line Tools
win_get_url:
  url:
https://dl.google.com/android/repository/commandlinetools-win-6200805_latest.zip
  dest: C:/Programms
```

⁸⁶ <https://developer.android.com/studio>

8. Έπειτα, πραγματοποιήθηκε εξαγωγή του αρχείου αυτού στη διαδρομή «C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-tools», η οποία δημιουργήθηκε από το αμέσως επόμενο «task» με τη χρήση της ενότητας λογισμικού (module) «win_unzip» που περιγράφηκε αναλυτικά στο βήμα 6, και η αναφερόμενη εργασία (task) έχει ως εξής:

```
- name: Extract Command Line Tools
```

```
win_unzip:
```

```
src: C:/Programms/commandlinetools-win-6200805_latest.zip
```

```
dest: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-tools
```

```
creates: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-tools
```

9. Ο φάκελος «tools» που εξήχθη στη διαδρομή αυτή, μετονομάστηκε σε «latest». Η διαδικασία αυτή χρειάζεται να υλοποιηθεί, διότι διαφορετικά, δεν λειτουργεί ο διαχειριστής «SDK» (SDK Manager) που θα χρησιμοποιηθεί για την εγκατάσταση των εργαλείων «SDK» (SDK Tools), ο οποίος δεν εγκαθίσταται μαζί με το λογισμικό «Android Studio», όπως προαναφέρθηκε⁸⁷. Για την εκτέλεση των παραπάνω, χρησιμοποιήθηκαν το «module» «win_shell»⁸⁸, με το οποίο εκτελείται στον υπολογιστή – στόχο, μια εντολή «cmd-let» στη διεπαφή «PowerShell» του υπολογιστή – στόχου. Η εντολή αυτή πραγματοποιεί τη μετονομασία του παραπάνω φακέλου. Σημειώνεται ότι χρησιμοποιήθηκε το «module» «win_shell», διότι δεν υπάρχει κάποιο «module» που να μπορεί να πραγματοποιεί μετονομασία σε κάποιο αρχείο ή/και φάκελο⁸⁹. Το «task» έχει ως εξής:

⁸⁷ <https://stackoverflow.com/questions/5199811/android-sdk-manager-wont-open>
<https://stackoverflow.com/questions/60440509/android-command-line-tools-sdkmanager-always-shows-warning-could-not-create-se>

⁸⁸ https://docs.ansible.com/ansible/latest/modules/win_shell_module.html

⁸⁹ <https://stackoverflow.com/questions/47774130/how-to-rename-move-a-file-on-a-remote-windows-host-with-ansible>
https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html

```
- name: Rename folder tools
```

```
win_shell: 'Rename-Item -Path "C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-  
tools/tools" -NewName "latest"'
```

10. Ύστερα, δημιουργήθηκε η μεταβλητή περιβάλλοντος (Environment Variable) «JAVA_HOME» στην οποία ως τιμή δίδεται η διαδρομή όπου βρίσκεται η «Java», δηλαδή «C:\Program Files\Java\jdk-14.0.1.». Αυτή η διαδικασία είναι απαραίτητη προκειμένου η εικονική μηχανή με λειτουργικό σύστημα «Windows 10», να «γνωρίζει» που βρίσκεται η «Java» για να μπορεί να τη χρησιμοποιήσει στις εφαρμογές που απαιτείται⁹⁰, όπως για παράδειγμα το «Android Studio» και τα εργαλεία «SDK» (SDK Tools) που εξετάζονται στη συγκεκριμένη μελέτη περίπτωσης. Η διαδικασία αυτή έγινε με τη χρήση του «module» «win_environment»⁹¹, το οποίο εφαρμόστηκε μέσω του αμέσως επόμενου «task»:

```
- name: Set environment variable JAVA_HOME for Java
```

```
win_environment:
```

```
state: present
```

```
name: JAVA_HOME
```

```
value: C:\Program Files\Java\jdk-14.0.1
```

```
level: machine
```

Όπου:

- **Win_environment:** Ενότητα λογισμικού (module) η οποία χρησιμοποιείται για την τροποποίηση και δημιουργία μεταβλητών περιβάλλοντος (environmental variables) σε λειτουργικά συστήματα «Windows». Το «module» αυτό χρησιμοποιεί το «.net» περιβάλλον των «Windows» για τον ορισμό ή την κατάργηση μεταβλητών

⁹⁰ <https://stackoverflow.com/questions/34770570/how-to-define-java-home-variable-for-android-studio>
<https://developer.android.com/studio/command-line/variables>

⁹¹ https://docs.ansible.com/ansible/latest/modules/win_environment_module.html

περιβάλλοντος και μπορεί να ορισθεί σε επίπεδο χρήστη, μηχανήματος ή διαδικασίας.

- **state**: Η μεταβλητή εκφράζει την ύπαρξη ή όχι της μεταβλητής περιβάλλοντος προς ορισμό. Στην προκειμένη περίπτωση δίδεται η τιμή «present» για να διασφαλιστεί ότι έχει ρυθμιστεί η μεταβλητή περιβάλλοντος.
- **name**: Το όνομα της μεταβλητής περιβάλλοντος.
- **value**: Η τιμή που θα δοθεί στην προαναφερόμενη μεταβλητή περιβάλλοντος.
- **Level**: Το «level» ορίστηκε σε «machine» και όχι σε «user», προκειμένου να μπορεί κάθε προφίλ χρήστη που χρησιμοποιεί τον εν λόγω ηλεκτρονικό υπολογιστή, να μπορεί να χρησιμοποιεί τη συγκεκριμένη μεταβλητή περιβάλλοντος και να μην απαιτείται εκ νέου η δημιουργία της σε κάθε προφίλ χρήστη.

11. Στη συνέχεια, προκειμένου να υλοποιηθεί η εγκατάσταση των εργαλείων «SDK» (SDK Tools), θα πρέπει να ορισθεί άλλη μία μεταβλητή περιβάλλοντος, η «ANDROID_HOME», η οποία έχει ως τιμή τη διαδρομή που θα βρίσκονται τα εργαλεία «SDK» (SDK Tools), ήτοι «C:\Users\%username%\AppData\Local\Android\Sdk». Χρησιμοποιήθηκε αυτή η διαδρομή, διότι τα εργαλεία «SDK» (SDK Tools) εγκαθίστανται στη συγκεκριμένη διαδρομή από προεπιλογή⁹². Αυτό έγινε με το ίδιο «module» «win_environment» που περιγράφηκε στο βήμα 10 και διαμορφώνεται ως εξής:

```
- name: Set environment variable ANDROID_HOME sdktools installation
  win_environment:
    state: present
    name: ANDROID_HOME
    value: C:\Users\winuser\AppData\Local\Android\Sdk
    level: machine
```

⁹² <https://developer.android.com/studio/command-line/variables>

<https://stackoverflow.com/questions/19375098/android-studio-path-android-home-not-able-to-set>

Επιπλέον, αυτές οι μεταβλητές θα πρέπει να καταχωρηθούν και στη μεταβλητή περιβάλλοντος «PATH» των «Windows», ως «%JAVA_HOME%/bin» και ως «%ANDROID_HOME%»⁹³ αντίστοιχα. Η παραπάνω διαδικασία αποσκοπεί στο να χρησιμοποιείται από τον υπολογιστή - στόχο η συγκεκριμένη διαδρομή όταν χρειάζεται τη γλώσσα «Java». Η διαδικασία αυτή πραγματοποιήθηκε με τη χρήση του «module win_path». Όλη η παραπάνω διαδικασία περιγράφεται στην παρακάτω εργασία (task):

```
- name: Set PATH environment variable for all users
  win_path:
    elements:
      - '%ANDROID_HOME%'
      - '%JAVA_HOME%\bin'
    scope: machine
```

Όπου:

- **win_path:** Δεν χρησιμοποιήθηκε το «module» «win_environment» πάλι, διότι αυτό το «module» δεν επιτρέπει καταχωρήσεις, μεταβολές ή ο,τιδήποτε σχετίζεται με τη μεταβλητή περιβάλλοντος, «PATH» των λειτουργικών συστημάτων «Windows»⁹⁴.
- **elements:** Μεταβλητή με την οποία εκφράζεται ποια στοιχεία θα προστεθούν στην μεταβλητή περιβάλλοντος, «PATH». Τα στοιχεία αυτά θα καταχωρισθούν με την ίδια σειρά που θα ορισθούν στη μεταβλητή αυτή.

⁹³ https://www.ntu.edu.sg/home/ehchua/programming/android/Android_HowTo.html

⁹⁴ https://docs.ansible.com/ansible/latest/modules/win_path_module.html

- **Scope:** Να σημειωθεί και εδώ ότι στο «scope» δόθηκε η τιμή «machine» προκειμένου οι παραπάνω μεταβλητές περιβάλλοντος να είναι διαθέσιμες σε όλα τα προφίλ χρηστών που χρησιμοποιούν το συγκεκριμένο υπολογιστή - στόχο.

12. Μετέπειτα, εκτελέστηκε το αρχείο «sdkmanager.bat» αρχείο από τη διαδρομή «C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-tools\latest\bin». Με την εκτέλεση του αρχείου αυτού, πραγματοποιείται η εγκατάσταση των εργαλείων «SDK» (SDK Tools) που είναι επιθυμητά. Αρχικά, εκτελέστηκε η εντολή «for(\$i=0;\$i -lt 30;\$i++) { \$response += "γ`n"; \$response | .\sdkmanager.bat –licenses», προκειμένου να επιτευχθεί η αυτοματοποιημένη αποδοχή των αδειών χρήσης των εργαλείων που θα εγκατασταθούν⁹⁵.

Η εντολή αυτή ωστόσο, προκειμένου να εκτελεστεί σωστά, θα πρέπει να εκτελεστεί στη διεπαφή «PowerShell» με δικαιώματα διαχειριστή, και για να επιτευχθεί αυτό χρησιμοποιήθηκαν οι μεταβλητές «ansible_become» «ansible_become_method» «ansible_become_user» και «ansible_become_password», οι οποίες «καλούν» τη μέθοδο «run as» και προσδιορίζουν τα διαπιστευτήρια του τοπικού διαχειριστή των «Windows». Στην ουσία, με αυτές τις μεταβλητές, είναι σαν να γίνεται στη διεπαφή «Powershell» της εικονικής μηχανής με λειτουργικό σύστημα «Windows» η επιλογή της δυνατότητας «εκτέλεση ως διαχειριστής».

Η μεταβλητή «chdir» εκφράζει την εντολή «cd» στη διεπαφή «Powershell», με την οποία μεταβαίνει στην διαδρομή που λαμβάνει ως όρισμα. Όλη η διαδικασία αυτή πραγματοποιήθηκε με το «module win_shell»⁹⁶.

```
- name: Accept all licenses for sdk tools via sdk manager
```

⁹⁵ <https://stackoverflow.com/questions/38096225/automatically-accept-all-sdk-licences?page=1&tab=votes#tab-top>

⁹⁶ https://docs.ansible.com/ansible/latest/modules/win_shell_module.html

```

win_shell: 'for($i=0;$i -lt 30;$i++) { $response += "y`n"};
$response | .\sdkmanager.bat --licenses'
args:
    chdir: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-
tools/latest/bin
vars:
    ansible_become: yes
    ansible_become_method: runas
    ansible_become_user: admin
    ansible_become_password: "{{ vault_ansible_become_password
}}"

```

13. Τέλος, εκτελέστηκε η εντολή «sdkmanager.bat "emulator" "build-tools;29.0.3" "platforms;android-29" "platform-tools" "patcher;v4" "sources;android-29"», με την οποία πραγματοποιήθηκε η εγκατάσταση των εργαλείων που βρίσκονται εντός εισαγωγικών ("""). Τα εργαλεία αυτά είναι τα απαραίτητα συστατικά «SDK» (SDK Components)⁹⁷, που χρειάζεται το λογισμικό «Android Studio». Για την διεκπεραίωση αυτής της διαδικασίας χρησιμοποιήθηκε το «module» «win_command»⁹⁸, διότι το αρχείο «sdkmanager.bat», δεν αποτελεί αρχείο το οποίο μπορεί να εκτελεστεί σε «PowerShell», αλλά σε γραμμή εντολών “Windows” (Command Prompt- CMD). Επιπλέον, η ορθή του εκτέλεση απαιτεί δικαιώματα διαχειριστή και αναβαθμισμένο κατ’ επέκταση τερματικό γραμμής εντολών. Για το λόγο αυτό χρησιμοποιήθηκαν οι μεταβλητές οι οποίες χρησιμοποιήθηκαν και στο προηγούμενο «task» που εκτελέστηκε στο βήμα 12⁹⁹.

⁹⁷ <https://developer.android.com/studio/intro/update> <https://developer.android.com/studio/releases/sdk-tools>

⁹⁸ https://docs.ansible.com/ansible/latest/modules/win_command_module.html

⁹⁹ <https://developer.android.com/studio/command-line/sdkmanager>
<https://tool.oschina.net/uploads/apidocs/android/sdk/adding-components.html>
https://docs.ansible.com/ansible/latest/modules/win_command_module.html

```
- name: Install platform via sdk manager
  win_command: 'sdkmanager.bat "emulator" "build-tools;29.0.3"
"platforms;android-29" "platform-tools" "patcher;v4" "sources;android-
29"'
  args:
    chdir: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-
tools/latest/bin
  vars:
    ansible_become: yes
    ansible_become_method: runas
    ansible_become_user: admin
    ansible_become_password: "{{ vault_ansible_become_password
}}"
```

Το ολοκληρωμένο «playbook» που εκτελεί τις παραπάνω αναφερόμενες διαδικασίες παρουσιάζεται στο Παράρτημα Α΄ της παρούσας διπλωματικής εργασίας.

Το παραπάνω «playbook» εκτελείται με την εντολή στο τερματικό του μηχανήματος ελέγχου του «Ansible» (Ansible control machine).

```
ansible-playbook -i inventory.yml playbooks/AndroidStudio.yml -vvvvvv -ask-vault-pass
```

Όπου:

- **Παράμετρος -i:** Με τη χρήση της παραμέτρου αυτής ορίζεται το αρχείο «inventory» που χρησιμοποιείται.
- **Παράμετρος -vvvvv:** Με τη χρήση της παραμέτρου αυτής (verbose), εμφανίζονται λεπτομέρειες για το κάθε «task» του «playbook» που εκτελείται, οι οποίες μπορούν να χρησιμοποιηθούν για τη μελέτη και αντιμετώπιση σφαλμάτων.
- **Παράμετρος -ask-vault-pass:** Με τη χρήση της παραμέτρου αυτής ζητείται ο κωδικός που ορίστηκε για το «Vault». Χωρίς αυτή την παράμετρο η εκτέλεση του «Playbook» δεν είναι δυνατή, διότι στο αρχείο «Playbook» εμπεριέχονται οι μεταβλητές που κρυπτογραφήθηκαν με τη χρήση του «Ansible Vault».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

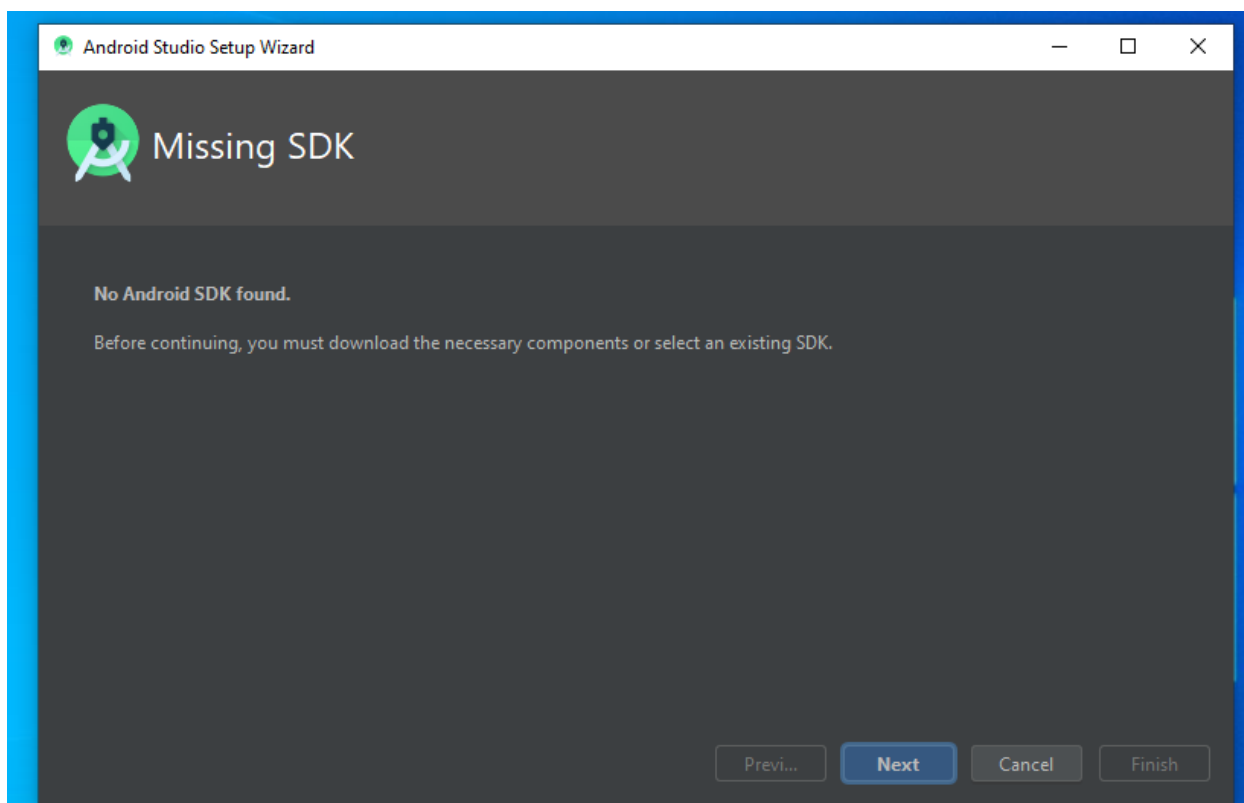
Ελισάβετ Αραμπατζή

Αφού εκτελεστεί το «playbook» επιτυχώς, προκύπτει το εξής αποτέλεσμα:

```
PLAY RECAP *****
192.168.162.4      : ok=13  changed=11  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

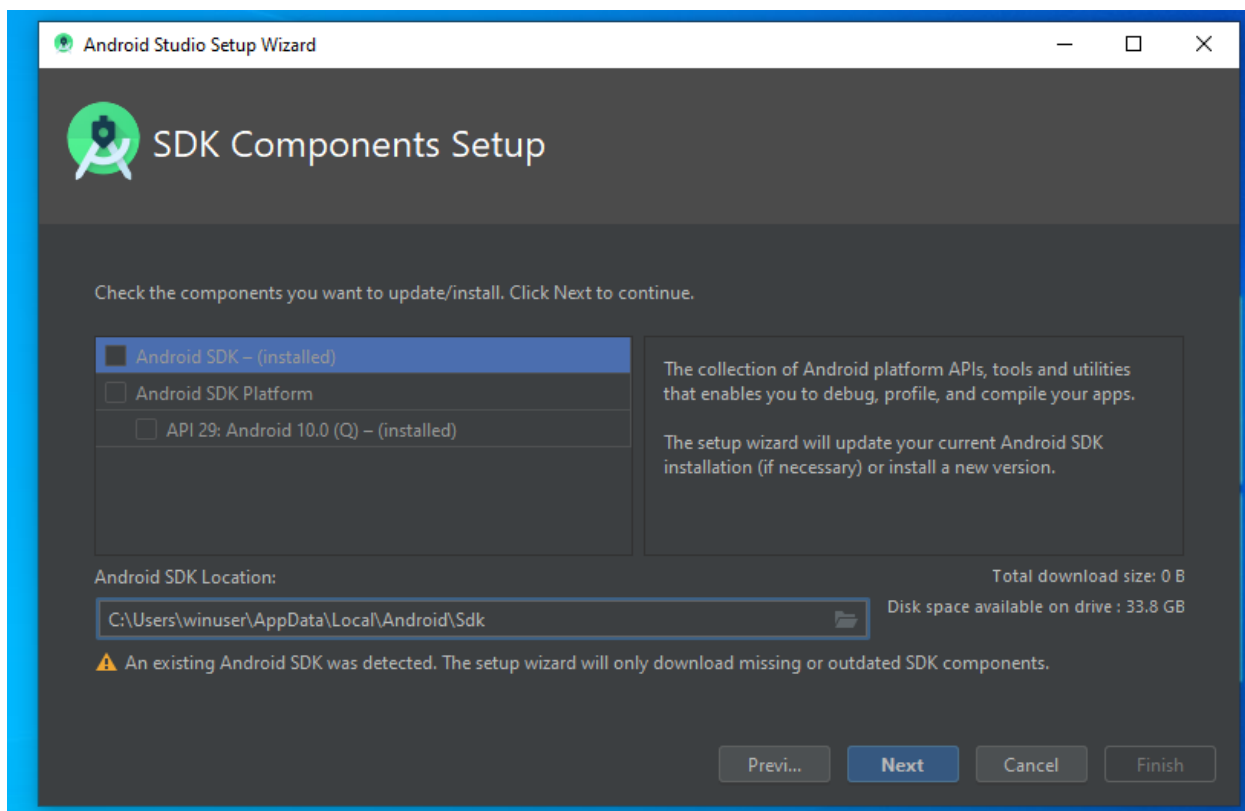
Εικόνα 32: Συνολικά αποτελέσματα από την ολοκλήρωση της εκτέλεσης «Playbook».

Από την πλευρά της εικονικής μηχανής με λειτουργικό σύστημα «Windows 10» εκτελώντας το λογισμικό «Android Studio», ως τελικός χρήστης, εμφανίζονται οι εξής οθόνες, μόνο εφόσον έχει εκτελεσθεί επιτυχώς το «playbook»:



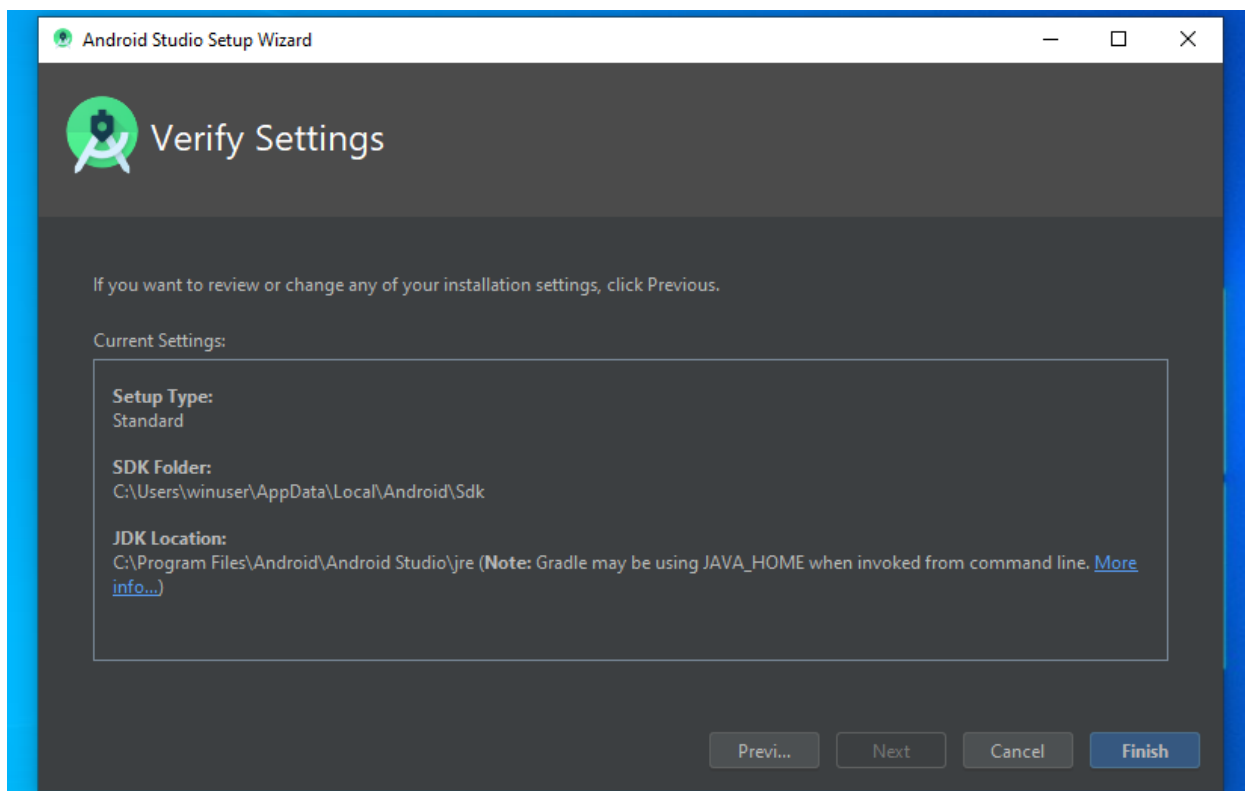
Εικόνα 33: Αναζήτηση των εργαλείων «SDK» (SDK Tools).

Επιλέγοντας «Next» εμφανίζεται η παρακάτω οθόνη, όπου παρατηρείται ότι το «Android Studio» έχει εντοπίσει τα «SDK Tools» που εγκαταστάθηκαν στην τοποθεσία που είχε ορισθεί στο «Playbook» προηγουμένως:

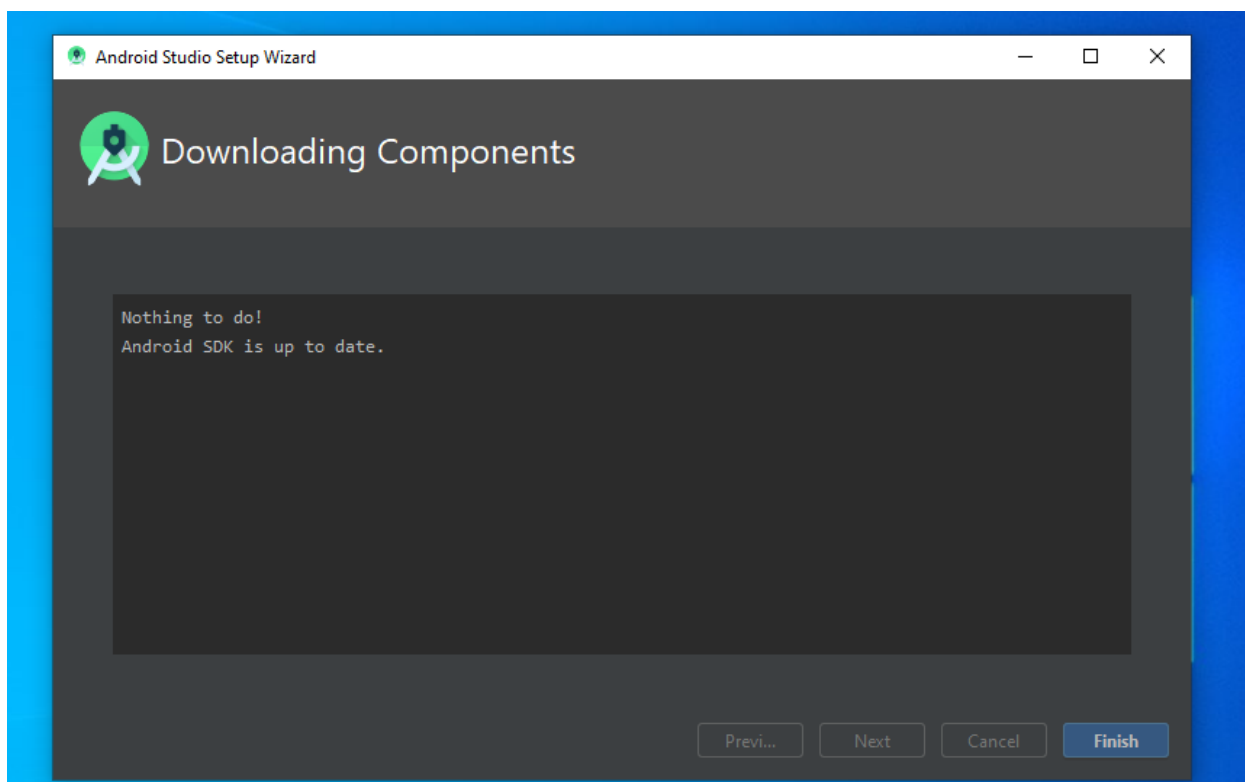


Εικόνα 34: Εντοπισμός «SDK Tools».

Στην παρακάτω οθόνη εμφανίζεται μια σύνοψη των ρυθμίσεων του λογισμικού «Android Studio». Επιλέγοντας «Finish» και προκύπτει το εξής αποτέλεσμα:



Εικόνα 35: Σύνοψη Ρυθμίσεων Λογισμικού «Android Studio» στον τελικό χρήστη (end user).



Εικόνα 36: Επιτυχής Αυτοματοποιημένη Εγκατάσταση του «Android Studio» και των «SDK Tools».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

4.2 Εισαγωγή Ηλεκτρονικού Υπολογιστή με λειτουργικό σύστημα «Windows» σε δίκτυο τομέα «Windows Domain Network»

Μία από τις επαναλαμβανόμενες διαδικασίες που απαιτείται να εκτελέσει ένας διαχειριστής μιας υποδομής πληροφορικής σε ένα παραγωγικό περιβάλλον, είναι η εισαγωγή ενός υπολογιστή σε κάποιο δίκτυο τομέα (domain network), όπως ίσως και η εξαγωγή του από αυτό.

Αρχικά, εξετάστηκε η εισαγωγή (join) του υπολογιστή στο δίκτυο τομέα «Windows» (Windows domain network) και στην επόμενη [παράγραφο 4.3](#) θα εξετασθεί η εξαγωγή (unjoin) από το εν λόγω δίκτυο τομέα.

Προκειμένου να επιτευχθεί αυτό, χρησιμοποιήθηκε το παρακάτω «playbook», το οποίο ονομάστηκε «domainin.yml». Το «playbook» αυτό δημιουργήθηκε όπως και το προηγούμενο της [παραγράφου 4.1](#), χρησιμοποιώντας τον κειμενογράφο «Vim», στη διαδρομή «/home/debuseradmin/playbooks», στη εικονική μηχανή που περιέχει το «Ansible» (Ansible Control Machine) και οι διαδικασίες που εφαρμόστηκαν, έχουν ως εξής:

1. Αρχικά στο πρώτο «task», με τη χρήση του «module» «win_domain_membership»¹⁰⁰ γίνεται η εισαγωγή (join) της εικονικής μηχανής με λειτουργικό σύστημα «Windows» στο δίκτυο τομέα (domain network) του εικονικού μηχανήματος με λειτουργικό «Windows server 2016» όπως περιγράφεται στη συνέχεια:

```
---  
  
- name: Join windows host to windows domain  
  hosts: winhost  
  
  tasks:  
  - name: Join to Domain test.local  
    win_domain_membership:
```

¹⁰⁰https://docs.ansible.com/ansible/latest/modules/win_domain_membership_module.html


```
dns_domain_name: test.local
hostname: win10pro
domain_admin_user: test.local\joindomain
domain_admin_password: "{{ vault_domain_admin_password }}"
state: domain
register: domain_state
```

Όπου:

- **win domain membership**: Ενότητα λογισμικού (module), η οποία δύναται να εισάγει κάποιον Ηλεκτρονικό Υπολογιστή (Windows host) σε ένα δίκτυο τομέα περιβάλλοντος «Windows» (Windows domain network).
- **dns domain name**: Όταν η μεταβλητή «state», η οποία αναλύεται παρακάτω, λαμβάνει την τιμή «domain», η μεταβλητή «dns_domain_name» λαμβάνει ως τιμή το πλήρες όνομα του δικτύου τομέα (domain name) στο οποίο πρόκειται να εισαχθεί ο υπολογιστής – στόχος (host). Στην εξεταζόμενη περίπτωση δίδεται η τιμή «test.local» που είναι και το όνομα του δικτύου τομέα του περιβάλλοντος δοκιμών και εξέτασης των μελετών περίπτωσης.
- **hostname**: Μεταβλητή η οποία εκφράζει το επιθυμητό όνομα που θα λάβει ο υπολογιστής – στόχος αφού εισαχθεί στο δίκτυο τομέα (domain network). Στη συγκεκριμένη περίπτωση, δόθηκε η τιμή «win10pro».
- **domain admin user**: Μεταβλητή στην οποία ορίζονται τα διαπιστευτήρια του χρήστη, ο οποίος διαθέτει τα απαραίτητα δικαιώματα για να εισάγει τον υπολογιστή – στόχο (host) στο δίκτυο τομέα. Στη εν λόγω μελέτη, ορίστηκε η τιμή «test.local\joindomain».
- **domain admin password**: Ο κωδικός πρόσβασης του παραπάνω χρήστη, κρυπτογραφημένος με τη χρήση του «Ansible Vault».

- **state:** Μεταβλητή που ορίζεται εάν ο υπολογιστής - στόχος θα εισαχθεί σε δίκτυο τομέα (domain) ή σε ομάδα εργασίας (workgroup). Στη συγκεκριμένη περίπτωση ορίσθηκε η τιμή «domain».
- **register:** Με τη μεταβλητή αυτή, ορίζεται η μεταβλητή «domain_state» η οποία θα χρησιμοποιηθεί στο αμέσως επόμενο «task» και δηλώνει εάν υπήρξε αλλαγή στην κατάσταση του υπολογιστή – στόχου σε σχέση με το δίκτυο – τομέα (domain network), προκειμένου να «αποφασισθεί» εάν θα γίνει επανεκκίνηση ή όχι.

2. Προκειμένου να ολοκληρωθεί η εισαγωγή ενός υπολογιστή με λειτουργικό σύστημα «Windows» (Windows host) σε ένα δίκτυο τομέα «Windows» απαιτείται επανεκκίνηση¹⁰¹. Για το σκοπό αυτό, χρησιμοποιήθηκε η ενότητα λογισμικού (module) «win_reboot»¹⁰², όπως περιγράφεται στη συνέχεια:

```
- name: Reboot winhost
  win_reboot:
    vars:
      ansible_winrm_operation_timeout_sec: 200
      ansible_winrm_read_timeout_sec: 600
      ansible_winrm_transport: ntlm
      ansible_user: test.local\joindomain
      ansible_password: "{{ vault_domain_admin_password }}"
    when: domain_state.reboot_required
```

Όπου:

- **win_reboot:** Ενότητα λογισμικού (module) που υλοποιεί επανεκκίνηση υπολογιστών με λειτουργικά συστήματα «Windows». Στη συγκεκριμένη περίπτωση, χρησιμοποιείται για την ολοκλήρωση της εισαγωγής (join) του υπολογιστή – στόχου στο δίκτυο τομέα (domain).

¹⁰¹ <https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/join-a-computer-to-a-domain>

¹⁰² https://docs.ansible.com/ansible/latest/modules/win_reboot_module.html

- Στη συνέχεια, επειδή η υπηρεσία «WinRM» είναι αρκετά ευαίσθητο σε δικτυακές μεταβολές¹⁰³ και προκειμένου να μην διακόπτεται η επικοινωνία μεταξύ του μηχανήματος ελέγχου που περιέχει το «Ansible» (Ansible Control Machine) μεταβλήθηκαν οι τιμές των εξής μεταβλητών:
 - **ansible_winrm_operation_timeout_sec**: Η μεταβλητή αυτή αφορά στην αύξηση του προεπιλεγμένου χρονικού ορίου για λειτουργίες της υπηρεσίας «WinRM». Το «Ansible» χρησιμοποιεί την τιμή 20 από προεπιλογή, η οποία μεταβλήθηκε σε 200.
 - **ansible_winrm_read_timeout_sec**: Η μεταβλητή αυτή αφορά στο χρονικό όριο ανάγνωσης της υπηρεσίας «WinRM». Το «Ansible» χρησιμοποιεί την τιμή 30 από προεπιλογή. Η μεταβλητή είναι χρήσιμη εάν υπάρχουν προβλήματα όπου διακόπτεται το δίκτυο και εξακολουθούν να εμφανίζονται σφάλματα χρονικού ορίου. Προκειμένου να μην προκύπτουν τέτοια ζητήματα, η τιμή αυξήθηκε σε 300.
- Επιπλέον, επαναπροσδιορίστηκαν οι μεταβλητές (vars) «**ansible_winrm_transport**», «**ansible_user**» και «**ansible_password**», οι οποίες στο αρχείο «inventory» έχουν άλλες τιμές, πλην της μεταβλητής «**ansible_winrm_transport**». Αυτό συμβαίνει διότι από τη στιγμή που εισήχθη ο υπολογιστής στο δίκτυο τομέα (domain network), θα χρειαστεί να χρησιμοποιηθούν διαφορετικά διαπιστευτήρια (credentials), τα οποία πλέον σχετίζονται με το δίκτυο τομέα (domain network), για να εκτελεστούν μέσα από την υπηρεσία «WinRM» τα υπόλοιπα «tasks». Συνεπώς, η χρήση των μεταβλητών αυτών κατ' αυτόν τον τρόπο, οδηγεί στην επικοινωνία του μηχανήματος ελέγχου του «Ansible» (Ansible control machine), με την εικονική μηχανή με λειτουργικό σύστημα «Windows».

Το παραπάνω «playbook» εκτελέστηκε με τη χρήση της εξής εντολής στο τερματικό:

```
ansible-playbook -i inventory.ini playbooks/domainin.yml -vvvvvv --ask-vault-pass
```

¹⁰³ <https://github.com/ansible/ansible/issues/23320>

Όπου:

- **Παράμετρος -i:** Με τη χρήση της παραμέτρου αυτής ορίζεται το αρχείο «inventory» που χρησιμοποιείται.
- **Παράμετρος -vvvvv:** Με τη χρήση της παραμέτρου αυτής (verbose), εμφανίζονται λεπτομέρειες για το κάθε «task» του «playbook» που εκτελείται, οι οποίες μπορούν να χρησιμοποιηθούν για τη μελέτη και αντιμετώπιση σφαλμάτων.
- **Παράμετρος -ask-vault-pass:** Με τη χρήση της παραμέτρου αυτής ζητείται ο κωδικός που ορίστηκε για το «Vault». Χωρίς αυτή την παράμετρο η εκτέλεση του «playbook» δεν είναι δυνατή, διότι στο αρχείο «playbook» εμπεριέχονται οι μεταβλητές που κρυπτογραφήθηκαν με τη χρήση του «Ansible Vault».

Αποτέλεσμα της επιτυχούς εκτέλεσης του παραπάνω «playbook» φαίνεται στην παρακάτω εικόνα:

```
changed: [192.168.162.3] => {
  "changed": true,
  "elapsed": 47,
  "rebooted": true
}
META: ran handlers
META: ran handlers

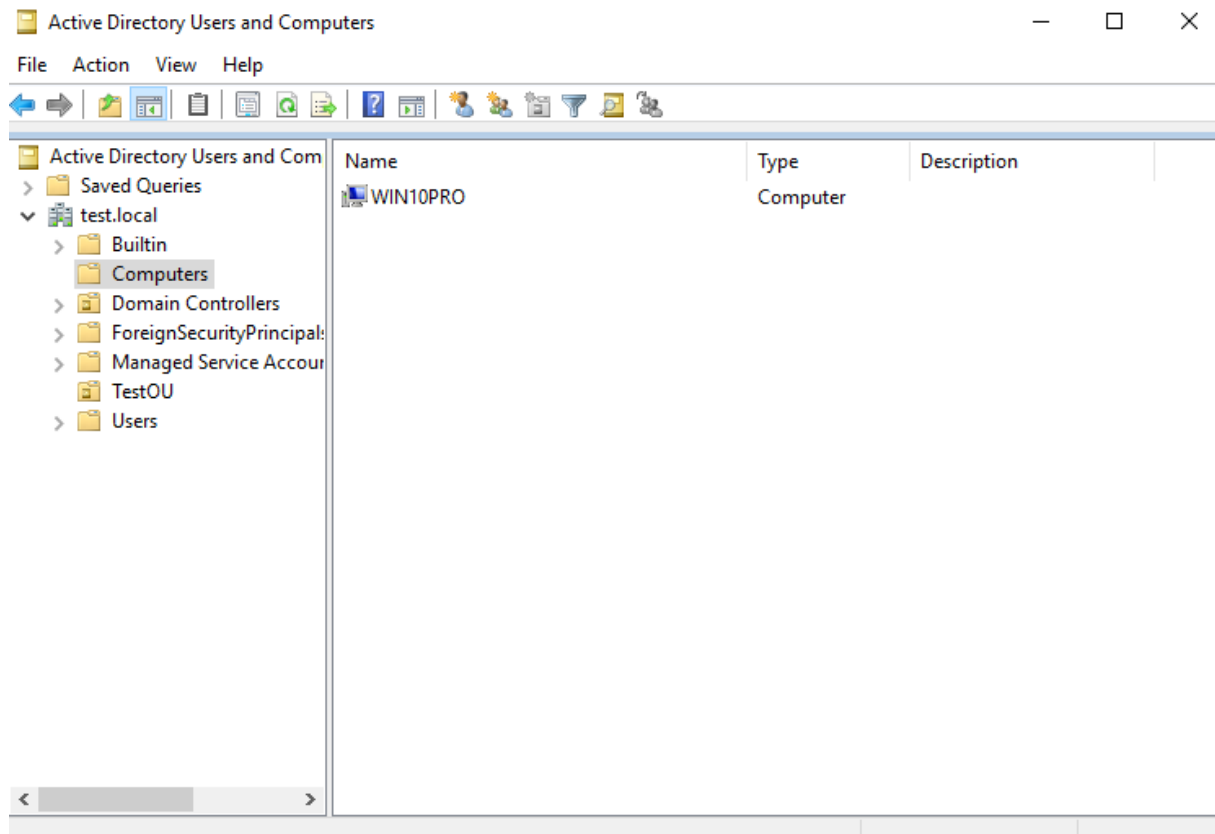
PLAY RECAP *****
192.168.162.3      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Εικόνα 37: Επιτυχής εκτέλεση «Playbook» εισαγωγής υπολογιστή στο δίκτυο τομέα (domain network).

Επιπλέον, από την πλευρά της εικονικής μηχανής με λειτουργικό σύστημα «Windows Server», παρατηρείται η επιτυχής εκτέλεση της διαδικασίας, διότι η εικονική μηχανή με λειτουργικό σύστημα «Windows 10» εισήχθη στην προεπιλεγμένη Οργανωτική Μονάδα (Organizational Unit – OU), με ονομασία «Computers». Θα μπορούσε να οριστεί και η εισαγωγή της σε διαφορετική Οργανωτική Μονάδα (Organizational Unit – OU), αλλά λόγω του ότι η διαδικασία αυτή υλοποιείται σε περιβάλλον δοκιμών, προτιμήθηκε η συγκεκριμένη Οργανωτική Μονάδα (Organizational Unit – OU).

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή



Εικόνα 38: Active Directory Windows Domain Controller.

Εάν ο υπολογιστής παρέμενε στο δίκτυο τομέα (domain network), θα ήταν πιο σωστό να τροποποιηθεί το αρχείο «inventory», και να χρησιμοποιηθεί λογαριασμός με δικαιώματα διαχειριστικά στο δίκτυο τομέα (domain network), και η επικοινωνία της υπηρεσίας «WinRM» να ορισθεί με βάση το πρωτόκολλο «kerberos», και όχι με το πρωτόκολλο «NTLM» που χρησιμοποιείται τώρα. Το πρωτόκολλο αυθεντικοποίησης «NTLM - New Technology Lan Manager) είναι ένας παλαιότερος μηχανισμός ελέγχου ταυτότητας που χρησιμοποιείται από τη «Microsoft» και μπορεί να υποστηρίξει τόσο τοπικούς όσο και λογαριασμούς σε δίκτυα τομέα (domain networks). Το «NTLM» είναι ενεργοποιημένο από προεπιλογή στην υπηρεσία «WinRM», οπότε δεν απαιτείται εγκατάσταση πριν τη χρήση του. Αν και τα δύο πρωτόκολλα αποτελούν τρόπο επικοινωνίας με κάποιον «domain controller», πιο ασφαλές είναι το πρωτόκολλο «Kerberos», καθώς υποστηρίζει λειτουργίες, όπως είναι η ανάθεση διαπιστευτηρίων (credential delegation) και η κρυπτογράφηση μηνυμάτων μέσω πρωτοκόλλου «HTTPS». Άλλωστε αποτελεί μία από τις πιο ασφαλείς επιλογές που είναι διαθέσιμη μέσω της

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

υπηρεσίας «WinRM». Απαιτεί, επιπλέον, παραμετροποίηση και στο μηχανήμα ελέγχου του «Ansible» (Ansible control machine), προκειμένου να λειτουργεί και να επικοινωνεί και αυτό στο πλαίσιο του δικτύου τομέα (domain network), ακόμη και αν δεν εισαχθεί κανονικά σε αυτό¹⁰⁴.

Το ολοκληρωμένο «playbook» που εκτελεί τις παραπάνω αναφερόμενες διαδικασίες παρουσιάζεται στο [Παράρτημα Β΄](#) της παρούσας διπλωματικής εργασίας.

4.3 Εξαγωγή Ηλεκτρονικού Υπολογιστή με λειτουργικό σύστημα «Windows» από δίκτυο τομέα «Windows Domain Network»

Προκειμένου να πραγματοποιηθεί η εξαγωγή της εικονικής μηχανής με λειτουργικό σύστημα «Windows» από το δίκτυο τομέα (domain network), χρησιμοποιήθηκε ένα «playbook» το οποίο ονομάστηκε «domainout.yml». Στο «playbook» αυτό χρησιμοποιήθηκαν ακριβώς τα ίδια «modules» που χρησιμοποιήθηκαν και κατά τη δημιουργία του «playbook» που αφορούσε την εισαγωγή (join) του υπολογιστή – στόχου (host) στο δίκτυο τομέα (domain network). Στη συγκεκριμένη περίπτωση, ωστόσο, υπάρχουν ορισμένες διαφοροποιήσεις οι οποίες παρουσιάζονται στη συνέχεια και έχουν ως εξής:

1. Αρχικά, για την εκτέλεση του αναφερόμενου «Playbook», επαναπροσδιορίστηκαν οι μεταβλητές «ansible_user» και «ansible_password». Στις μεταβλητές «**ansible_user**» και «**ansible_password**» ορίστηκαν τα διαπιστευτήρια (credentials) του χρήστη «test.local\joindomain». Ο χρήστης αυτός αποτελεί χρήστη του δικτύου τομέα και διαθέτει τα κατάλληλα δικαιώματα για την εισαγωγή / εξαγωγή στο δίκτυο τομέα (domain network), όπως προαναφέρθηκε και στην [παράγραφο 4.2](#). Αυτό έγινε διότι θεωρήθηκε ότι ο υπολογιστής – στόχος (host) είναι ήδη μέλος του δικτύου τομέα και συνεπώς, ο χρήστης «admin» που έχει οριστεί στο αρχείο «inventory» δεν μπορεί πλέον να χρησιμοποιηθεί για την εδραίωση της επικοινωνίας του «Ansible» με το λειτουργικό σύστημα «Windows» μέσω της υπηρεσίας «winRM» (WinRM Connection Pipeline).

¹⁰⁴ https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html#kerberos

```

---
- name: Unjoin windows host from windows domain
  hosts: winhost

  vars:
    ansible_user: test.local\joindomain
    ansible_password: "{{ vault_domain_admin_password }}"
    ansible_winrm_transport: ntlm

  tasks:
    - name: Unjoin from Domain test.local
      win_domain_membership:
        workgroup_name: WINWORKGROUP
        domain_admin_user: test.local\joindomain
        domain_admin_password: "{{ vault_domain_admin_password }}"
        state: workgroup

```

Οι υπόλοιπες μεταβλητές που χρησιμοποιούνται στο «task» που περιγράφει την εξαγωγή του υπολογιστή – στόχου (host) έχουν ως εξής:

- **workgroup name:** Μεταβλητή στην οποία εκφράζεται το όνομα της ομάδας (workgroup) που θα βρίσκεται ο υπολογιστής – στόχος (host) αφού εξαχθεί από το δίκτυο τομέα (domain network). Στη συγκεκριμένη περίπτωση ορίστηκε σε «WINWORKGROUP».
- **domain admin user:** Μεταβλητή στην οποία ορίζονται τα διαπιστευτήρια του χρήστη, ο οποίος διαθέτει τα απαραίτητα δικαιώματα για να εισάγει τον υπολογιστή – στόχο (host) στο δίκτυο τομέα. Στη εν λόγω μελέτη, ορίστηκε η τιμή «test.local\joindomain».

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

- **domain admin password:** Ο κωδικός πρόσβασης του παραπάνω χρήστη, κρυπτογραφημένος με τη χρήση του «Ansible Vault».
- **state:** Η μεταβλητή αυτή ορίζει εάν ο υπολογιστής – στόχος (host) εισαχθεί σε δίκτυο τομέα (domain) ή σε ομάδα εργασίας (workgroup). Στη συγκεκριμένη περίπτωση ορίστηκε η τιμή «workgroup».
- Τέλος, επαναπροσδιορίστηκαν και σε αυτό το σημείο οι μεταβλητές «**ansible_user**» και «**ansible_password**» ορίστηκαν τα διαπιστευτήρια (credentials) του χρήστη «test.local\joindomain».

2. Προκειμένου να ολοκληρωθεί η εξαγωγή ενός υπολογιστή «Windows» (Windows host) από ένα δίκτυο τομέα «Windows» απαιτείται επανεκκίνηση¹⁰⁵. Για το σκοπό αυτό δημιουργήθηκε ένα «task» για την επανεκκίνηση χρησιμοποιώντας την ενότητα λογισμικού «win_reboot», όπως και κατά την εξέταση της εισαγωγής στο δίκτυο τομέα (domain network), στην [παράγραφο 4.2](#). Το «task» αυτό υλοποιείται αμέσως μετά το «task» που αφορά στην εργασία εξαγωγή από το δίκτυο τομέα και έχει ως εξής:

```
- name: Reboot winhost
  win_reboot:
  vars:
    ansible_winrm_operation_timeout_sec: 200
    ansible_winrm_read_timeout_sec: 600
    ansible_winrm_transport: ntlm
    ansible_user: win10pro\admin
    ansible_password: "{{ vault_ansible_password }}"
```

Για την εκτέλεση αυτού του «task», η μόνη διαφοροποίηση από το όμοιο «task» στην [παράγραφο 4.2](#), είναι ότι οι μεταβλητές «**ansible_user**» και «**ansible_password**», επαναπροσδιορίστηκαν χρησιμοποιώντας τα διαπιστευτήρια του χρήστη «admin» με

¹⁰⁵ <https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/join-a-computer-to-a-domain>

δικαιώματα τοπικού διαχειριστή, ο οποίος είναι ορισμένος και στο αρχείο «inventory». Αυτό γίνεται διότι ο υπολογιστής – στόχος δεν είναι πλέον στο δίκτυο τομέα μετά την εκτέλεση του πρώτου «task» στο εν λόγω «playbook», και συνεπώς εάν δεν πραγματοποιηθούν οι εν λόγω αλλαγές δεν θα μπορεί να εκτελεστεί το «task» που αφορά στην επανεκκίνηση.

Το παραπάνω «playbook» εκτελέστηκε με τη χρήση της εξής εντολής στο τερματικό:

```
ansible-playbook -i inventory.ini playbooks/domainout.yml -vvvvvv --ask-vault-pass
```

Όπου:

- **Παράμετρος -i:** Με τη χρήση της παραμέτρου αυτής ορίζεται το αρχείο «inventory» που χρησιμοποιείται.
- **Παράμετρος -vvvvv:** Με τη χρήση της παραμέτρου αυτής (verbose), εμφανίζονται λεπτομέρειες για το κάθε «task» του «playbook» που εκτελείται, οι οποίες μπορούν να χρησιμοποιηθούν για την μελέτη και αντιμετώπιση σφαλμάτων.
- **Παράμετρος --ask-vault-pass:** Με τη χρήση της παραμέτρου αυτής ζητείται ο κωδικός που ορίστηκε για το «Vault». Χωρίς αυτή την παράμετρο η εκτέλεση του «playbook» δεν είναι δυνατή, διότι στο αρχείο «playbook» εμπεριέχονται οι μεταβλητές που κρυπτογραφήθηκαν με τη χρήση του «Ansible Vault».

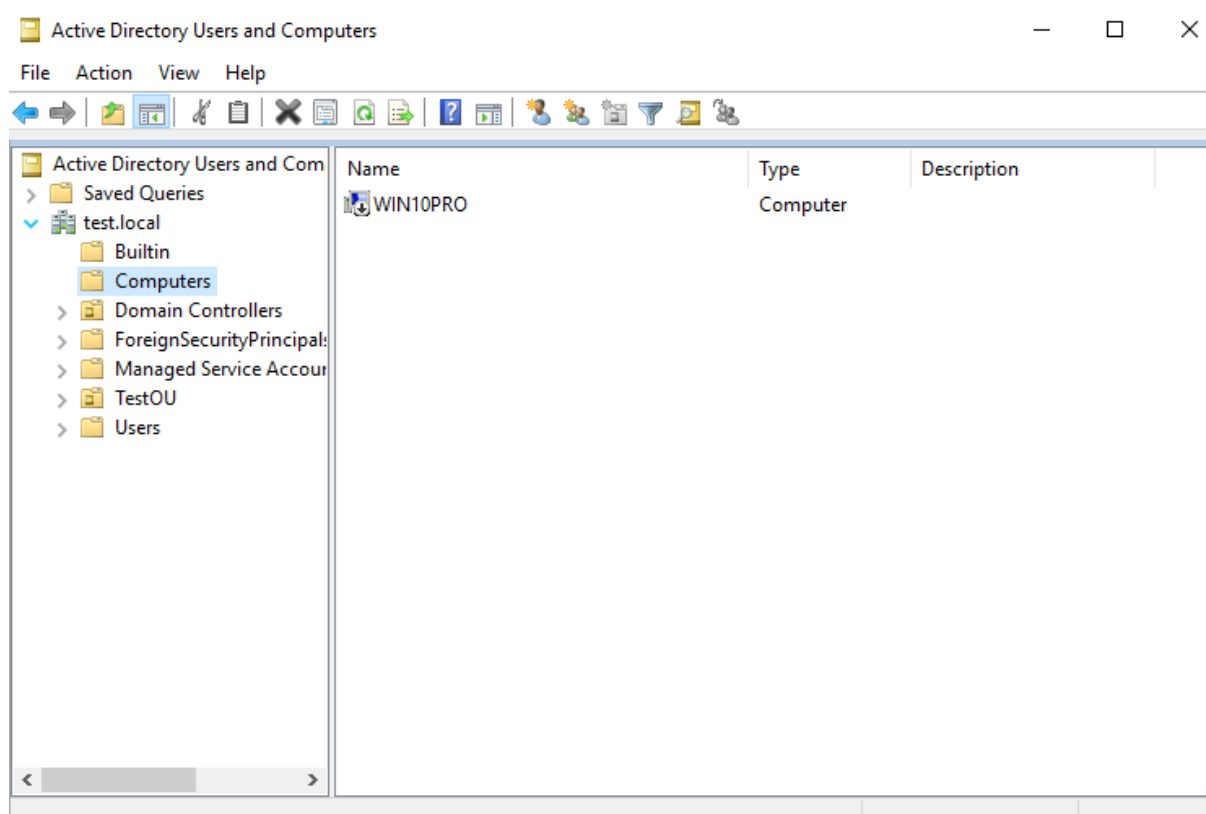
Αποτέλεσμα της επιτυχούς εκτέλεσης του παραπάνω «playbook» φαίνεται στην παρακάτω εικόνα:

```
changed: [192.168.162.3] => {
  "changed": true,
  "elapsed": 59,
  "rebooted": true
}
META: ran handlers
META: ran handlers

PLAY RECAP *****
192.168.162.3      : ok=3    changed=2    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

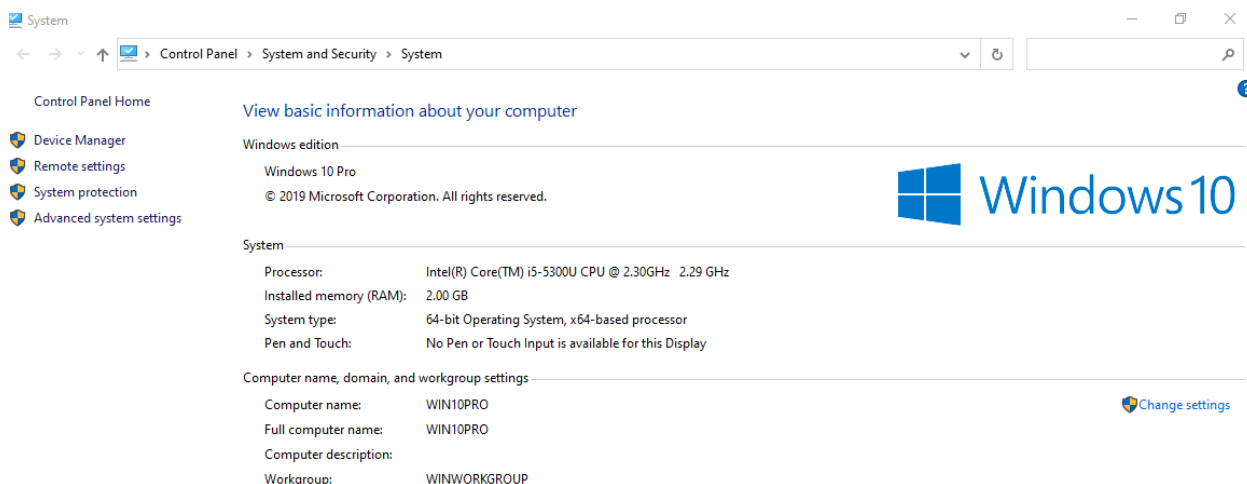
Εικόνα 39: Επιτυχής εκτέλεση «Playbook» εξαγωγής υπολογιστή από δίκτυο τομέα (domain network).

Επιπλέον, από την πλευρά της εικονικής μηχανής με λειτουργικό σύστημα «Windows Server», παρατηρείται η επιτυχής εκτέλεση της διαδικασίας, διότι η εικονική μηχανή με λειτουργικό σύστημα «Windows 10» εμφανίζεται πλέον ως «ανενεργή» (disabled) στην προεπιλεγμένη Οργανωτική Μονάδα (Organizational Unit – OU), με ονομασία «Computers». Θα μπορούσε να οριστεί και η εισαγωγή του σε διαφορετική Οργανωτική Μονάδα (Organizational Unit – OU), αλλά λόγω του ότι η διαδικασία αυτή υλοποιείται σε περιβάλλον δοκιμών, προτιμήθηκε η συγκεκριμένη Οργανωτική Μονάδα (Organizational Unit – OU).



Εικόνα 40: Απενεργοποίηση υπολογιστή που εξήχθη από το δίκτυο τομέα στο Active Directory στον Domain Controller Server.

Επιπλέον, η επιτυχής εκτέλεση της όλης διαδικασίας γίνεται αντιληπτή και στον υπολογιστή – στόχο, ο οποίος πλέον βρίσκεται στην ομάδα εργασίας η οποία ονομάστηκε «WINWORKGROUP».



Εικόνα 41: Υπολογιστής σε ομάδα εργασίας των λειτουργικών συστημάτων «Windows».

4.4 Εγκατάσταση λογισμικού προσομοίωσης «Simul8»

Στη συνέχεια, εξετάστηκε η εγκατάσταση του λογισμικού προσομοίωσης «Simul8» στην εικονική μηχανή με λειτουργικό σύστημα «Windows 10». Ωστόσο, η συγκεκριμένη εγκατάσταση δεν κατέστη εφικτή, διότι το αρχείο που παρασχέθηκε διαθέτει μαθητική άδεια λογισμικού (Student License). Αυτό σημαίνει σύμφωνα με τον επίσημο ιστότοπο που παρέχει το αναφερόμενο λογισμικό, ότι οι εκδόσεις που απευθύνονται σε μαθητικό προσωπικό, όπως επίσης και οι δοκιμαστικές (trial) εκδόσεις, δεν διαθέτουν στο εκτελέσιμο αρχείο τους διακόπτες (switches) ή παραμέτρους (arguments) που θα μπορούσαν να χρησιμοποιηθούν για την αυτοματοποιημένη εγκατάσταση (unattended installation) του λογισμικού. Οι εκδόσεις που μπορούν να χρησιμοποιηθούν για την εγκατάσταση με αυτοματοποιημένο τρόπο, είναι οι εκδόσεις «Corporate» και «Network»¹⁰⁶.

Ωστόσο, αναφέρεται ότι σε περίπτωση που ήταν διαθέσιμο το κατάλληλο λογισμικό με την άδεια χρήσης του, η ενδεικτική εκτέλεση της διαδικασίας του εν λόγω λογισμικού θα μπορούσε

¹⁰⁶ <https://support.simul8.com/silent-installs/>

να γίνει με την χρήση των ενοτήτων λογισμικού (modules) «win_shell» και «win_package», τα οποία αναλύθηκαν εκτενώς στην [παράγραφο 4.1](#).

1^{ος} Τρόπος

Ενδεικτικά παρουσιάζεται ο τρόπος με τη χρήση της ενότητας λογισμικού (module) «win_shell», όπου το «playbook» θα είχε ως εξής:

```
---
- name: install Simul8 Software
  hosts: winhost
  vars:
    ansible_winrm_transport: ntlm
    ansible_user: test.local\joindomain
    ansible_password: "{{ vault_domain_admin_password }}"

  tasks:
    - name: Install Simul8
      win_shell: /s LIC:"1959-2368-3548" DIR:"C:\Program Files"
      args:
        chdir: \\testsrv\Shares\Simul8
```

Αρχικά, επειδή θεωρήθηκε ότι ο υπολογιστής – στόχος αποτελεί μέλος του δικτύου τομέα (domain network), επαναπροσδιορίστηκαν οι μεταβλητές «**ansible_user**» και «**ansible_password**» στις οποίες ορίστηκαν τα διαπιστευτήρια (credentials) ενός χρήστη του δικτύου, που διαθέτει τα κατάλληλα διαχειριστικά δικαιώματα και μέσω του οποίου θα εδραιωθεί η επικοινωνία μεταξύ του «Ansible» και του υπολογιστή - στόχου.

Για την εγκατάσταση του λογισμικού «Simul8» αυτοματοποιημένα (unattended installation), απαιτείται η χρήση των εξής παραμέτρων (arguments):

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

- **Παράμετρος /s:** Το λογισμικό «Simul8» ανήκει στην κατηγορία πακέτων λογισμικού «Wise»¹⁰⁷ και απαιτείται η χρήση της συγκεκριμένης παραμέτρου για την αυτοματοποιημένη εγκατάσταση.
- **Παράμετρος LIC:"Licence Number":** Με τη χρήση της παραμέτρου αυτής επιτυγχάνεται η εισαγωγή του αριθμού της άδειας χρήσης και της αποδοχής αυτής. Χωρίς αυτή την παράμετρο, δεν θα υλοποιηθεί η εγκατάσταση.
- **Παράμετρος DIR:"Directory path":** Με τη χρήση της μεταβλητής αυτής, υλοποιείται η εγκατάσταση του εν λόγω λογισμικού στην επιθυμητή διαδρομή. Εάν δεν δοθεί αυτή η παράμετρος, η εγκατάσταση θα γίνει στον προεπιλεγμένο φάκελο από τα «Windows» στη διαδρομή που βρίσκεται ο φάκελος «Program Files».

Σημειώνεται ότι, οι παράμετροι «LIC» και «DIR» αφορούν μόνο στο λογισμικό «Simul8» και δεν εφαρμόζεται η χρήση τους σε κάποιο άλλο πακέτο λογισμικού¹⁰⁸.

Επιπλέον, επιλέχθηκε η ενότητα λογισμικού «win_shell» και όχι η «win_command», διότι η διεπαφή «PowerShell» στην οποία εκτελούνται απευθείας εντολές με τη χρήση του «win_shell», επιτρέπει την αλλαγή φακέλου (change directory) σε μια δικτυακή διαδρομή (UNC path), όπως αυτή που βρίσκεται το εκτελέσιμο αρχείου του πακέτου λογισμικού. Αντιθέτως, το «cmd» στο οποίο εκτελούνται απευθείας εντολές με τη χρήση του «win_command», επιτρέπει την παραπάνω δυνατότητα μόνο σε μια τοπική διαδρομή (local path).

2^{ος} Τρόπος

Στη συνέχεια παρατίθεται και το ενδεικτικό «Playbook» στο οποίο χρησιμοποιείται η ενότητα λογισμικού «win_package», χωρίς να χρησιμοποιηθούν κάποιες διαφορετικές παράμετροι.

```
---
- name: install Simul8 Software
```

¹⁰⁷ <http://unattended.sourceforge.net/installers.php>

¹⁰⁸ <https://www.simul8.com/support/help/doku.php?id=troubleshoot:install:cmd>
<https://support.simul8.com/silent-installs/>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

```
hosts: winhost
vars:
    ansible_winrm_transport: ntlm
    ansible_user: test.local\joindomain
    ansible_password: "{{ vault_domain_admin_password }}"

tasks:
- name: Install Simul8
  win_package:
    path: \\testsrv\Shares\Simul8\StudentS8SETUP.EXE
    product_id: simul8
    arguments: /s LIC:"1959-2368-3548" DIR:"C:\Program Files"
```

Κεφάλαιο 5^ο: Συμπεράσματα και Μελλοντικές Επεκτάσεις

5.1 Συμπεράσματα

Ολοκληρώνοντας την εκπόνηση της παρούσας διπλωματικής εργασίας, προέκυψαν χρήσιμα και εποικοδομητικά συμπεράσματα ως προς τις δυνατότητες που παρέχει η διαδικασία της αυτοματοποιημένης παραμετροποίησης υποδομών πληροφορικής με τη χρήση του εργαλείου «Ansible» τα οποία αναλύονται στη συνέχεια.

Ένα από τα θετικότερα στοιχεία που διαπιστώθηκαν αποτελεί ότι το «Ansible» μπορεί να χαρακτηριστεί ως εργαλείο «όλα σε ένα». Όλα τα βήματα αυτοματισμού μιας υποδομής μπορούν να υλοποιηθούν με το «Ansible». Αυτό οφείλεται στο γεγονός ότι το «Ansible» αναπτύχθηκε με τη γλώσσα προγραμματισμού «Python», η οποία αποτελεί μια γλώσσα προγραμματισμού γενικής χρήσης και επιπλέον διαθέτει μια τεράστια υποστηρικτική κοινότητα. Αυτό σημαίνει ότι μπορούν να χρησιμοποιηθούν, τα χιλιάδες υπάρχοντα πακέτα (Python Packages) από την κοινότητα της «Python» και οποιοσδήποτε να δημιουργήσει τις δικές του ενότητες λογισμικού (modules). Κατά συνέπεια, σε ένα περιβάλλον αυτοματισμού «μηδενικής ωριμότητας» (zero automation maturity) μιας επιχείρησης ή ενός οργανισμού, δηλαδή περιβάλλοντα τα οποία δεν διαθέτουν αυτοματισμούς ή διαθέτουν ελάχιστα στοιχεία αυτοματισμών, το «Ansible» μπορεί να συμβάλλει σημαντικά έτσι ώστε το εκάστοτε τμήμα Πληροφορικής τους να επιτύχει την αυτοματοποίηση της της υποδομής με τον γρηγορότερο δυνατό τρόπο.

Επιπλέον, αναφορικά με τη διαχείριση της διαμόρφωσης (Configuration Management) μιας υποδομής πληροφορικής με τη χρήση του «Ansible», ο κώδικας που συγγράφεται, ο κύκλος ζωής της εκάστοτε παραμετροποίησης που απαιτείται να γίνει, και οι οποιοσδήποτε αλλαγές, πραγματοποιούνται μέσω της διαμόρφωσης των αρχείων «inventory», με τη χρήση των «Ansible Playbooks» και των ρόλων του «Ansible» (Ansible roles). Έτσι, καθίσταται δυνατή η επίτευξη της επιθυμητής κατάστασης στον υπολογιστή – στόχο, καθώς και ο ουσιαστικός έλεγχος των τυχών αδυναμιών στα «tasks» που θα εκτελεστούν.

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Επιπρόσθετα, το «Ansible» διαθέτει πολύ χαμηλή καμπύλη εκμάθησης. Η σύνταξη και οι ροές εργασίας είναι κατανοητές, καθιστώντας εύκολη την εκμάθηση για νέους χρήστες. Η δηλωτική γλώσσα «YAML» που χρησιμοποιείται για την σύνταξη των «Playbooks», δεν απαιτεί ιδιαίτερες γνώσεις συγγραφής κώδικα, γεγονός που καθιστά το «Ansible» ως εργαλείο το οποίο θα μπορούσε να χρησιμοποιηθεί από διαχειριστές πληροφοριακών συστημάτων σχεδόν κάθε γνωστικού υποβάθρου. Επιπρόσθετα, η χρήση του συμβάλλει τα μέγιστα στην τυποποίηση των διαδικασιών πληροφορικής, εξασφαλίζοντας την απρόσκοπτη λειτουργία μιας υποδομής πληροφορικής. Η «YAML», αποτελεί ένα φιλικό προς το χρήστη πρότυπο δηλωτικής γλώσσας που χρησιμοποιείται ευρέως και από άλλα εργαλεία και είναι εύκολα κατανοητό.

Επίσης, το «Ansible» διαθέτει εύκολη εγκατάσταση και αρχική διαμόρφωση. Σε πολύ σύντομο χρονικό διάστημα, είναι δυνατή η εγκατάσταση του εργαλείου, η ρύθμιση παραμέτρων του και η εκτέλεση εντολών “ad-hoc” είτε προς εξυπηρετητές (Servers) είτε προς υπολογιστές – πελάτες (hosts) που στόχο έχουν την αυτοματοποιημένη παραμετροποίησή τους. Οι μόνες, ουσιαστικά απαιτήσεις, είναι η εγκατάσταση της «Python» στα λειτουργικά συστήματα «Linux», η οποία ούτως ή άλλως υπάρχει στις περισσότερες διανομές «Linux» και της υπηρεσίας «WinRM» στα λειτουργικά συστήματα «Windows», η οποία είναι ενσωματωμένη στα λειτουργικά συστήματα «Windows» και απλά χρειάζεται κάποιες απλές παραμετροποιήσεις. Από τη στιγμή που το «ping» στους υπολογιστές – στόχους (hosts) μέσω του «Ansible» είναι επιτυχημένο, μπορεί να ξεκινήσει αμέσως η αυτοματοποίηση της υποδομής.

Τέλος, επιπλέον θετικό στοιχείο που διαπιστώθηκε, είναι η ευελιξία που παρέχει το εργαλείο «Ansible». Η μη ύπαρξη κάποιου «πράκτορα» (agentless) και η αλληλεπίδρασή του με τα «Windows» μέσα από μια υπηρεσία, αυτή του «WinRM», καθιστά το «Ansible» πολύ εύκολο στην παραμετροποίησή του. Τα «Playbooks» ωθούν (push) τις επιθυμητές ρυθμίσεις στους υπολογιστές – στόχους που ορίζονται στο εκάστοτε αρχείο «inventory». Τα «Playbooks» μπορούν να εκτελεστούν και «ad hoc», μέσω της γραμμής εντολών, χωρίς να απαιτούν ορισμούς σε αρχεία. Η χρήση του μοντέλου χωρίς «πράκτορες» (agentless), καθιστά πιο εύκολη και γρήγορη τόσο την επικοινωνία, όσο και τις διάφορες προσαρμογές και διαμορφώσεις από την εφαρμογή του μοντέλου πελάτη – εξυπηρετητή (client – server model). Μέσω της χρήσης των «modules» του «Ansible», εκτελέστηκαν με επιτυχία ενέργειες όπως η αλλαγή κωδικού

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

πρόσβασης διαχειριστή, η ενημέρωση εξυπηρετητή, η επανεκκίνηση υπηρεσιών και υπολογιστών, η εισαγωγή (join) και η εξαγωγή (unjoin) υπολογιστή σε δίκτυο τομέα (domain network), η αντιγραφή αρχείων, η λήψη αρχείων, η συμπίεση και η αποσυμπίεση αρχείων, η αλλαγή και καταχώριση μεταβλητών περιβάλλοντος σε λειτουργικό σύστημα «Windows», καθώς και η εκτέλεση εντολών απευθείας στη διεπαφή «PowerShell» ενός υπολογιστή – στόχου (host).

Ωστόσο, κατά τη διαδικασία υλοποίησης των μελετών περίπτωσης, διαπιστώθηκαν και κάποια αρνητικά στοιχεία.

Ένα αρνητικό στοιχείο συνιστά η έλλειψη της έννοιας της κατάστασης. Το «Ansible» δεν παρακολουθεί τις εξαρτήσεις (dependencies) μεταξύ των εντολών ή/και των «modules» που εμπεριέχονται στις εργασίες (tasks) σε ένα «Playbook» και εκτελεί απλώς διαδοχικά τις εργασίες (tasks) από πάνω προς τα κάτω (top to bottom), και σταματά όταν ολοκληρωθούν όλες οι εργασίες ή όταν έστω μία από αυτές αποτύχει ή προκύψει οποιοδήποτε σφάλμα. Αποτέλεσμα αυτού είναι η αλλαγή της κατάστασης του υπολογιστή – στόχου (target host), σε ορισμένα σημεία και η αποτυχία επίτευξης της επιθυμητής κατάστασης.

Επιπλέον, η περιορισμένη υποστήριξη των «Windows», συνιστά σημαντικό μειονέκτημα. Το «Ansible» από την έκδοση «1.7» και έπειτα, υποστηρίζει και λειτουργικά συστήματα «Windows». Ωστόσο, δεν παρέχεται η δυνατότητα εξεύρεσης λύσης σε ό,τι αφορά τη χρήση του «Ansible» για την αυτοματοποιημένη παραμετροποίηση συστημάτων «Windows». Η πιο αξιόπιστη πηγή άντλησης πληροφοριών για την δημιουργία «Playbooks» τα οποία στοχεύουν σε «Windows» είναι ο επίσημος ιστότοπος του «Ansible»¹⁰⁹ στον οποίο εμπεριέχεται το σύνολο των «modules» που μπορούν να χρησιμοποιηθούν για την αυτοματοποιημένη παραμετροποίηση και διαχείριση κόμβων «Windows».

Συνεπώς, για την εξέταση του συνόλου των μελετών περίπτωσης της εν λόγω διπλωματικής εργασίας, δαπανήθηκε αρκετός χρόνος για τον πειραματισμό με έναν πολύ μεγάλο αριθμό «modules» που αφορούν στα λειτουργικά συστήματα «Windows», διότι δεν υφίσταται επαρκής

¹⁰⁹ https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

πληροφόρηση για τη χρήση και παραμετροποίηση του εκάστοτε «module» που αναφέρεται σε «Windows».

Το προαναφερόμενο ζήτημα κατά κύριο λόγο εντοπίστηκε κατά τη μελέτη εγκατάστασης πακέτων λογισμικού. Αυτό οφείλεται στο γεγονός ότι υπάρχουν αρκετά πακέτα λογισμικού για τα «Windows», όπως είναι τα πακέτα λογισμικού «EXE», «MSI» «InstallShield», «InstallShield with MSI», «Wise InstallMaster», «Inno Setup», «Nullsoft Scriptable Install System - NSIS», «Ghost installer», καθώς επίσης και τα «Microsoft hotfixes» και τα «Microsoft Windows Updates»¹¹⁰. Κάθε τύπος πακέτου λογισμικού διαθέτει και τις δικές του παραμέτρους επίτευξης αυτοματοποιημένης εγκατάστασης ή απεγκατάστασης, αποδοχής αδειών χρήσης κλπ, οι οποίες είναι απαραίτητο να ορισθούν στα «Ansible modules» που χρησιμοποιούνται για την εκτέλεση τέτοιων εργασιών ή ακόμη και σε «modules» τα οποία μπορούν να χρησιμοποιηθούν για την απομακρυσμένη απευθείας εκτέλεση εντολών στο «PowerShell» ενός «Windows target host», όπως τα «modules» «win_shell» και «win_command» που έχουν αναλυθεί.

Μία ακόμα αδυναμία του «Ansible» είναι έλλειψη «εμπειρίας» στο χώρο των επιχειρήσεων. Το εργαλείο δεν επιλέγεται ιδιαίτερα από μεγάλες επιχειρήσεις, σε σχέση με κάποιους ανταγωνιστές του, όπως «Puppet» και «Chef», με αποτέλεσμα η ελκυστικότητά του να μειώνεται. Παρόλο που η εταιρεία «Red Hat» που παρέχει το «Ansible» ισχυρίζεται ότι παρέχει διευρυμένες επιλογές υποστήριξης σε επίπεδο επιχείρησης, η περιορισμένη πρακτική εμπειρία αποδυναμώνει το επιχείρημά αυτό.

Επίσης, το «Ansible» συνιστά νεοεισερχόμενο εργαλείο στην αγορά, σε σχέση με τους πιο γνωστούς ανταγωνιστές του. Κατά συνέπεια, αυτό αυξάνει τις πιθανότητες ύπαρξης σφαλμάτων και ζητημάτων του λογισμικού που δεν έχουν ακόμα εντοπιστεί.

Σε κάθε περίπτωση, όμως, με τη χρήση του «Ansible», εκτελέστηκε μεγάλος αριθμός φαινομενικά απλών εργασιών, εφαρμόζοντας τις βέλτιστες δυνατές πρακτικές, δίνοντας πάντα προτεραιότητα σε εργασίες που προσθέτουν αξία και επιλύουν σημαντικά ζητήματα ή ικανοποιούν ουσιώδεις απαιτήσεις.

¹¹⁰ <http://unattended.sourceforge.net/installers.php>

Τα αποτελέσματα ήταν ιδιαίτερα ενθαρρυντικά και διαφαίνεται έντονα η επιτακτική ανάγκη αλλαγής πλεύσης των σύγχρονων διαχειριστών πληροφοριακών συστημάτων, στην εφαρμογή μεθόδων και εργαλείων που προωθούν την αυτοματοποίηση των διαδικασιών. Η λήψη, ωστόσο, μιας τέτοιας απόφασης για τη χρήση ενός τέτοιου εργαλείου, όπως είναι το «Ansible» μόνο θετικά αποτελέσματα θα είχε να προσδώσει σε έναν οργανισμό ή μια επιχείρηση, αυξάνοντας την παραγωγικότητα των εργαζομένων, προωθώντας τη συνεργασία και αυξάνοντας τα έσοδα τους.

5.2 Μελλοντικές Επεκτάσεις

Κατά την εκπόνηση της παρούσας διπλωματικής εργασίας, δεν εξετάστηκε το σύνολο των επιλογών και δυνατοτήτων που παρέχει το «Ansible», αφού κάτι τέτοιο θα ήταν εκτός του τιθέμενου ερευνητικού εύρους της. Για το λόγο αυτό, αξίζει να αναφερθούν κάποιες πιθανές μελλοντικές χρήσεις και ερευνητικές περιοχές όπου το εργαλείο «Ansible» και οι δυνατότητές του θα μπορούσαν να βρουν εφαρμογή.

Αρχικά, το εργαλείο «Ansible» μπορεί να χρησιμοποιηθεί και για την παραμετροποίηση «Windows server», σε περιβάλλοντα μεικτών λειτουργικών συστημάτων ή/και σε περιβάλλοντα όπου υφίστανται μόνο «Windows». Προκειμένου, ωστόσο, να επιτευχθεί αυτό, απαιτείται ειδικότερη παραμετροποίηση κατά την αρχική δημιουργία του μηχανήματος ελέγχου που περιέχει το «Ansible», η οποία δε διαφέρει ιδιαίτερα από αυτήν που αφορά ίσως έναν μόνο «host». Απαραίτητη προϋπόθεση για να μπορεί το «Ansible» να λειτουργήσει σε ένα «Windows domain/forest» και να μπορεί να χρησιμοποιηθεί και στην παραμετροποίηση «Windows Server», αποτελεί η εισαγωγή του στο «domain». Επιπρόσθετα, το «Ansible» μπορεί να χρησιμοποιηθεί και για την παραμετροποίηση «ESXi Server», ο οποίος φιλοξενεί υποδομή αποτελούμενη από εικονικά μηχανήματα «Windows».

Σε κάθε περίπτωση, τα στοιχεία που θα πρέπει να εισαχθούν αρχικά στο μηχανήμα ελέγχου που περιέχει το «Ansible» είναι διαφορετικά, έτσι ώστε να καταστεί εφικτή η επικοινωνία του εργαλείου με οποιοδήποτε κομμάτι μιας υποδομής, όπως η προαναφερόμενη, η οποία μπορεί να περιέχει μεικτά περιβάλλοντα λειτουργικών συστημάτων, που εξυπηρετούν όμως την ίδια

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

υποδομή. Τέτοιες υποδομές απαντώνται σε μεσαίους ή/και μεγάλους οργανισμούς, όπως για παράδειγμα, ένα πανεπιστήμιο, μια στρατιωτική μονάδα μεγάλου βεληνεκούς, ή μια εταιρεία κλπ.

Επιπρόσθετα, το «Ansible» αποτελεί μια απλοποιημένη αλλά ισχυρή μηχανή ενορχήστρωσης. Το «Ansible» ενσωματώνεται σε σχεδόν κάθε τομέα μιας υποδομής πληροφορικής. Επικεντρώνεται, επίσης, σε περιοχές ενδιαφέροντος όπου άλλα εργαλεία αφήνουν ανέγγιχτες, όπως ενδεικτικά οι συνεχείς ενημερώσεις στις εφαρμογές πολλαπλών επιπέδων σε ολόκληρη την υποδομή. Επιπλέον, το «Ansible» διαθέτει λειτουργικές μονάδες για «κοντέινερ» (π.χ. Docker), εικονικοποίηση (Virtualization) (π.χ. VMware, AWS, OpenStack, Azure, Ovirt) και μπορεί εύκολα να ενσωματωθεί με υφιστάμενες διαδικασίες ενός οργανισμού ή μιας επιχείρησης.

Τέλος, το «Ansible» διαθέτει επί του παρόντος περισσότερες από 1300 ενότητες λογισμικού (modules) σε διάφορους τομείς, που μπορούν να εφαρμοστούν σε όλο το φάσμα της υποδομής πληροφορικής και αφορούν σε εφαρμογές ιστού (web), βάσεις δεδομένων (Databases), παραμετροποίηση δικτύων (Network Automation), υπολογιστικό νέφος (cloud), περιβάλλοντα «συστάδων» δικτύων και υπολογιστικών συστημάτων (clusters), παρακολούθηση (monitoring), παραμετροποίηση λειτουργικών συστημάτων «Windows», παραμετροποίηση περιβαλλόντων αποθηκευτικού χώρου (storage environment) και πολλά περισσότερα.

Βιβλιογραφία

1. <https://searchitoperations.techtarget.com/definition/IT-automation>.
(προσπελάστηκε 04-05-2020)
https://en.wikipedia.org/wiki/IT_process_automation.
(προσπελάστηκε 04-05-2020)
<https://www.redhat.com/en/topics/automation/whats-it-automation>
(προσπελάστηκε 04-05-2020)
2. <https://www.vmware.com/topics/glossary/content/it-automation>
(προσπελάστηκε 03-05-2020)
<https://searchitoperations.techtarget.com/definition/IT-automation>
(προσπελάστηκε 04-05-2020)
3. <https://www.vmware.com/topics/glossary/content/it-automation>
(προσπελάστηκε 03-05-2020)
<https://searchitoperations.techtarget.com/definition/IT-automation>
(προσπελάστηκε 04-05-2020)
<https://www.redhat.com/en/topics/automation/whats-it-automation>
(προσπελάστηκε 04-05-2020)
4. <https://www.vmware.com/topics/glossary/content/it-automation>
(προσπελάστηκε 03-05-2020)
<https://searchitoperations.techtarget.com/definition/IT-automation>
(προσπελάστηκε 04-05-2020)
5. <https://www.redhat.com/en/resources/ansible-continuous-integration-delivery-whitepaper>
(προσπελάστηκε 04-05-2020)
<https://www.redhat.com/en/engage/automated-enterprise-ebook-20171107>
(ανακτήθηκε 04-05-2020)
<https://www.redhat.com/cms/managed-files/ma-network-automation-for-everyone-e-book-f19707wg-201910-en.pdf>
(ανακτήθηκε 04-05-2020)
6. <https://www.redhat.com/en/engage/automated-enterprise-ebook-20171107>
(ανακτήθηκε 04-05-2020)
<https://www.redhat.com/cms/managed-files/ma-network-automation-for-everyone-e-book-f19707wg-201910-en.pdf>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

(ανακτήθηκε 04-05-2020)

7. https://en.wikipedia.org/wiki/Software-defined_networking
(προσπελάστηκε 05-05-2020)
8. <https://www.marketsandmarkets.com/Market-Reports/change-configuration-management-market-103332706.html>
(προσπελάστηκε 05-05-2020)
9. <https://www.datanyze.com/market-share/configuration-management-313/Datanyze%20Universe?page=1>
(προσπελάστηκε 07-06-2020)
10. <https://www.datanyze.com/market-share/configuration-management--313/Greece>
(προσπελάστηκε 07-06-2020)
11. <https://cfengine.com/product/>
(προσπελάστηκε 04-06-2020)
<https://en.wikipedia.org/wiki/CFEngine>
(προσπελάστηκε 04-06-2020)
<https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 04-06-2020)
12. <https://northern.tech/products/cfengine>
(προσπελάστηκε 04-06-2020)
<https://cfengine.com/product/>
(προσπελάστηκε 04-06-2020)
13. <https://cfengine.com/product/what-is-cfengine/>
(προσπελάστηκε 04-06-2020)
14. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
15. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
16. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

(προσπελάστηκε 18-04-2020)

17. <https://www.oreilly.com/library/view/learning-cfengine-3/9781449334536/ch01.html>
(προσπελάστηκε 04-06-2020)
18. <https://www.oreilly.com/library/view/learning-cfengine-3/9781449334536/ch01.html>
(προσπελάστηκε 04-06-2020)
19. [https://en.wikipedia.org/wiki/Rudder_\(software\)](https://en.wikipedia.org/wiki/Rudder_(software))
(προσπελάστηκε 08-06-2020)
<https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
20. <https://www.rudder.io/pricing/subscription/>
(προσπελάστηκε 08-06-2020)
21. <https://docs.rudder.io/reference/6.0/index.html>
(ανακτήθηκε 08-06-2020)
22. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
23. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
24. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
25. [https://en.wikipedia.org/wiki/Puppet_\(company\)](https://en.wikipedia.org/wiki/Puppet_(company))
(προσπελάστηκε 07-06-2020)
<https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
26. <https://puppet.com/use-cases/windows-infrastructure-automation/>
(προσπελάστηκε 07-06-2020)
27. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

28. <https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
29. [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software))
(προσπελάστηκε 09-06-2020)
<https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
30. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
<https://www.chef.io/pricing/>
(προσπελάστηκε 09-06-2020)
31. [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software))
(προσπελάστηκε 09-06-2020)
<https://www.chef.io/it-automation/>
(προσπελάστηκε 09-06-2020)
32. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
33. [https://en.wikipedia.org/wiki/Chef_\(software\)](https://en.wikipedia.org/wiki/Chef_(software))
(προσπελάστηκε 09-06-2020)
<https://www.chef.io/why-chef/>
(προσπελάστηκε 09-06-2020)
34. [https://en.wikipedia.org/wiki/Salt_\(software\)](https://en.wikipedia.org/wiki/Salt_(software))
(προσπελάστηκε 12-06-2020)
35. <https://www.saltstack.com/>
(προσπελάστηκε 12-06-2020)
36. <https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
37. <https://docs.saltstack.com/en/latest/>
(προσπελάστηκε 12-06-2020)
38. [https://en.wikipedia.org/wiki/Salt_\(software\)](https://en.wikipedia.org/wiki/Salt_(software))

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

- (προσπελάστηκε 12-06-2020)
39. <https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
<https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
40. https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html
(ανακτήθηκε 12-03-2020)
41. https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html
(ανακτήθηκε 20-06-2020)
42. [https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))
(προσπελάστηκε 20-05-2020)
<https://www.ansible.com/overview/it-automation>
(προσπελάστηκε 20-05-2020)
43. <https://www.upguard.com/articles/configuration-management-tools>
(προσπελάστηκε 11-06-2020)
44. <https://www.ansible.com/overview/it-automation>
(προσπελάστηκε 20-05-2020)
<https://www.ansible.com/overview/how-ansible-works?intcmp=701f2000000h4RcAAI>
(προσπελάστηκε 20-05-2020)
[https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))
(προσπελάστηκε 20-05-2020)
<https://www.softwaretestinghelp.com/top-5-software-configuration-management-tools/>
(προσπελάστηκε 18-04-2020)
45. <https://www.ansible.com/products/tower>
(προσπελάστηκε 20-05-2020)
46. https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html
(ανακτήθηκε 15-06-2020)
47. <https://www.statista.com/statistics/218089/global-market-share-of-windows-7/#statisticContainer>
(προσπελάστηκε 07-06-2020)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

48. <https://www.ansible.com/>
(προσπελάστηκε 20-05-2020)
<https://www.ansible.com/overview/how-ansible-works>
(προσπελάστηκε 20-05-2020)
49. <https://www.vmware.com/products/workstation-pro.html>
(προσπελάστηκε 04-05-2020)
50. <https://el.wikipedia.org/wiki/Vim>
(προσπελάστηκε 15-05-2020)
<http://www.vim.org/>
(προσπελάστηκε 15-05-2020)
51. https://www.vmware.com/support/ws5/doc/ws_team_lan_requirements.html
(προσπελάστηκε 06-05-2020)
52. https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#selecting-an-ansible-version-to-install
(προσπελάστηκε 17-05-2020)
53. <https://www.ansible.com/integrations/infrastructure/windows>
(προσπελάστηκε 17-05-2020)
54. https://docs.ansible.com/ansible/latest/user_guide/windows_faq.html
(προσπελάστηκε 18-05-2020)
<https://books.google.gr/books?id=qvnpDwAAQBAJ&pg=PA9&lpg=PA9&dq=WSL+Ansible&source=bl&ots=RMM3NekMqM&sig=ACfU3U1L7mwy-guZPA5I76IPq6NxBJ9fbw&hl=en&sa=X&ved=2ahUKEwjghdrw4KzqAhWCGewKHSnaApUQ6AEwCXoECAgQAQ#v=onepage&q=WSL%20Ansible&f=false>
(ανακτήθηκε 18-05-2020)
55. https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#selecting-an-ansible-version-to-install
(προσπελάστηκε 17-05-2020)
<https://www.linuxtechi.com/install-ansible-automation-tool-debian10/>
(προσπελάστηκε 06-05-2020)
https://linuxhint.com/install_ansible_debian10/
(προσπελάστηκε 06-05-2020)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

56. <https://launchpad.net/~ansible/+archive/ubuntu/ansible>
(προσπελάστηκε 06-05-2020)
57. https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html
(προσπελάστηκε 15-06-2020)
<https://www.ansible.com/blog/connecting-to-a-windows-host>
(προσπελάστηκε 20-05-2020)
58. https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html
(προσπελάστηκε 20-05-2020)
59. https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#variables-and-vaults
(προσπελάστηκε 08-06-2020)
https://docs.ansible.com/ansible/latest/user_guide/playbooks_vault.html
(προσπελάστηκε 08-06-2020)
60. <https://www.digitalocean.com/community/tutorials/how-to-use-vault-to-protect-sensitive-ansible-data-on-ubuntu-16-04>
(προσπελάστηκε 08-06-2020)
61. [https://en.wikipedia.org/wiki/Jinja_\(template_engine\)](https://en.wikipedia.org/wiki/Jinja_(template_engine))
(προσπελάστηκε 19-06-2020)
<https://codeburst.io/jinja-2-explained-in-5-minutes-88548486834e>
(προσπελάστηκε 19-06-2020)
62. <https://github.com/PowerShell/Win32-OpenSSH/wiki/Install-Win32-OpenSSH>
(προσπελάστηκε 10-04-2020)
63. <https://www.ansible.com/blog/connecting-to-a-windows-host>
(προσπελάστηκε 20-05-2020)
https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html
(ανακτήθηκε 12-04-2020)
64. https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html
(ανακτήθηκε 12-04-2020)

65. https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html.
(ανακτήθηκε 12-04-2020)
<https://docs.ansible.com/ansible/latest/plugins/connection/winrm.html>.
(ανακτήθηκε 15-06-2020)
<https://adamtheautomator.com/winrm-https-ansible/>.
(προσπελάστηκε 17-04-2020)
<https://digitalist.global/talks/winrmansible/>
(προσπελάστηκε 15-06-2020)
66. <https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/enable-psremoting?view=powershell-7>
(ανακτήθηκε 15-04-2020)
67. <https://docs.microsoft.com/el-gr/previous-versions/windows/desktop/Policy/group-policy-objects?redirectedfrom=MSDN>.
(ανακτήθηκε 15-04-2020)
https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html#upgrading-powershell-and-net-framework
(ανακτήθηκε 15-04-2020)
68. https://docs.ansible.com/ansible/latest/user_guide/windows_setup.html#upgrading-powershell-and-net-framework
(ανακτήθηκε 15-04-2020)
69. <https://developer.android.com/studio/releases/sdk-tools>
(προσπελάστηκε 22-05-2020)
70. <https://developer.android.com/studio/intro/update#sdk-manager>
(προσπελάστηκε 22-05-2020)
71. <https://developer.android.com/studio>
(προσπελάστηκε 22-05-2020)
<https://developer.android.com/studio/releases/sdk-tools>
(προσπελάστηκε 22-05-2020)
72. https://www.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.html
(προσπελάστηκε 03-05-2020)

73. https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html
(ανακτήθηκε 25-04-2020)
74. <https://redirector.gvt1.com/edgedl/android/studio/install/3.6.2.0/android-studio-ide-192.6308749-windows.exe>
(προσπελάστηκε 08-05-2020)
75. https://docs.ansible.com/ansible/latest/modules/win_package_module.html
(ανακτήθηκε 23-05-2020)
76. <https://nsis.sourceforge.io/Docs/Chapter3.html>
(ανακτήθηκε 27-04-2020)
<https://superuser.com/questions/1432990/android-studio-nsis-error-while-installing>
(προσπελάστηκε 03-05-2020)
https://nsis.sourceforge.io/Main_Page
(προσπελάστηκε 03-05-2020)
77. <https://stackoverflow.com/questions/46038621/how-to-install-android-sdk-to-all-users-silently-on-windows>
(προσπελάστηκε 29-04-2020)
78. https://docs.ansible.com/ansible/latest/user_guide/become.html#become-and-windows
(ανακτήθηκε 29-04-2020)
79. https://docs.ansible.com/ansible/latest/user_guide/playbooks_error_handling.html
(ανακτήθηκε 29-04-2020)
80. <https://github.com/PSAppDeployToolkit/PSAppDeployToolkit/issues/365>
(προσπελάστηκε 29-04-2020)
81. https://docs.ansible.com/ansible/latest/user_guide/playbooks_error_handling.html#handlers-and-failure
(ανακτήθηκε 29-04-2020)
82. https://docs.ansible.com/ansible/latest/modules/win_file_module.html
(ανακτήθηκε 29-04-2020)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

83. https://download.java.net/java/GA/jdk14.0.1/664493ef4a6946b186ff29eb326336a2/7/GPL/openjdk-14.0.1_windows-x64_bin.zip
(προσπελάστηκε 08-05-2020)
84. https://docs.ansible.com/ansible/latest/modules/win_get_url_module.html
(ανακτήθηκε 29-04-2020)
85. https://docs.ansible.com/ansible/latest/modules/win_unzip_module.html
(ανακτήθηκε 29-04-2020)
86. <https://developer.android.com/studio>
(προσπελάστηκε 08-05-2020)
87. <https://stackoverflow.com/questions/5199811/android-sdk-manager-wont-open>
(προσπελάστηκε 15-04-2020)
<https://stackoverflow.com/questions/60440509/android-command-line-tools-sdkmanager-always-shows-warning-could-not-create-se>
(προσπελάστηκε 21-04-2020)
88. https://docs.ansible.com/ansible/latest/modules/win_shell_module.html
(ανακτήθηκε 29-04-2020)
89. <https://stackoverflow.com/questions/47774130/how-to-rename-move-a-file-on-a-remote-windows-host-with-ansible>
(προσπελάστηκε 21-04-2020)
https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html
(ανακτήθηκε 20-06-2020)
90. <https://stackoverflow.com/questions/34770570/how-to-define-java-home-variable-for-android-studio>
(προσπελάστηκε 02-05-2020)
<https://developer.android.com/studio/command-line/variables>
(προσπελάστηκε 02-05-2020)
91. https://docs.ansible.com/ansible/latest/modules/win_environment_module.html
(ανακτήθηκε 03-05-2020)

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

92. <https://developer.android.com/studio/command-line/variables>
(προσπελάστηκε 02-05-2020)
<https://stackoverflow.com/questions/19375098/android-studio-path-android-home-not-able-to-set>
(προσπελάστηκε 02-05-2020)
93. https://www.ntu.edu.sg/home/ehchua/programming/android/Android_HowTo.html
(προσπελάστηκε 09-05-2020)
94. https://docs.ansible.com/ansible/latest/modules/win_path_module.html
(ανακτήθηκε 12-05-2020)
95. <https://stackoverflow.com/questions/38096225/automatically-accept-all-sdk-licences?page=1&tab=votes#tab-top>
(προσπελάστηκε 26-04-2020)
96. https://docs.ansible.com/ansible/latest/modules/win_shell_module.html
(ανακτήθηκε 29-04-2020)
97. <https://developer.android.com/studio/intro/update>
(προσπελάστηκε 22-05-2020)
<https://developer.android.com/studio/releases/sdk-tools>
(προσπελάστηκε 22-05-2020)
98. https://docs.ansible.com/ansible/latest/modules/win_command_module.html
(ανακτήθηκε 03-05-2020)
99. <https://developer.android.com/studio/command-line/sdkmanager>
(προσπελάστηκε 30-04-2020)
<https://tool.oschina.net/uploads/apidocs/android/sdk/adding-components.html>
(προσπελάστηκε 30-04-2020)
https://docs.ansible.com/ansible/latest/modules/win_command_module.html
(ανακτήθηκε 03-05-2020)
100. https://docs.ansible.com/ansible/latest/modules/win_domain_membership_module.html
|
(ανακτήθηκε 17-06-2020)

101. <https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/join-a-computer-to-a-domain>
(ανακτήθηκε 17-06-2020)
102. https://docs.ansible.com/ansible/latest/modules/win_reboot_module.html
(ανακτήθηκε 17-06-2020)
103. <https://github.com/ansible/ansible/issues/23320>
(προσπελάστηκε 18-06-2020)
104. https://docs.ansible.com/ansible/latest/user_guide/windows_winrm.html#kerberos
(ανακτήθηκε 15-06-2020)
105. <https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/deployment/join-a-computer-to-a-domain>
(ανακτήθηκε 17-06-2020)
106. <https://support.simul8.com/silent-installs/>
(προσπελάστηκε 28-06-2020)
107. <http://unattended.sourceforge.net/installers.php>
(προσπελάστηκε 28-06-2020)
108. <https://www.simul8.com/support/help/doku.php?id=troubleshoot:install:cmd>
(προσπελάστηκε 28-06-2020)
<https://support.simul8.com/silent-installs/>
(προσπελάστηκε 28-06-2020)
109. https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html
(ανακτήθηκε 20-06-2020)
110. <http://unattended.sourceforge.net/installers.php>
(προσπελάστηκε 28-06-2020)

Παραρτήματα

Παράρτημα Α΄

«Playbook» για την εγκατάσταση του λογισμικού «Android Studio» και των εργαλείων «SDK» (SDK Tools)

```
---

- name: install Android Studio
  hosts: winhost

  tasks:
    - name: Install AS
      win_package:
        path:
https://redirector.gvt1.com/edgedl/android/studio/install/3.6.2.0/android-studio-ide-192.6308749-windows.exe
        product_id: android-studio
        arguments: /NCRC /S /IACCEPTTEULA
      ignore_errors: true
      #force_handlers: true
      failed_when: false
      changed_when: false

    - name: Create Directory for download tools
      win_file:
        path: C:/Programms
        state: directory

    - name: Download Java
      win_get_url:
```

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

```

        url:
https://download.java.net/java/GA/jdk14.0.1/664493ef4a6946b186ff29eb32
6336a2/7/GPL/openjdk-14.0.1_windows-x64_bin.zip
        dest: C:/Programms

- name: Extract Java to correct place
  win_unzip:
    src: C:/Programms/openjdk-14.0.1_windows-x64_bin.zip
    dest: C:/Program Files/Java
    creates: C:/Program Files/Java

- name: Download Command Line Tools
  win_get_url:
    url:
https://dl.google.com/android/repository/commandlinetools-win-
6200805_latest.zip
    dest: C:/Programms

- name: Extract Command Line Tools
  win_unzip:
    src: C:/Programms/commandlinetools-win-6200805_latest.zip
    dest: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-
tools
    creates:
C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-tools

- name: Rename folder tools
  win_shell: 'Rename-Item -Path
"C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-tools/tools" -
NewName "latest"'

- name: Set environment variable JAVA_HOME for Java
  win_environment:
    state: present
    name: JAVA_HOME

```

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

```

        value: C:\Program Files\Java\jdk-14.0.1
        level: machine

- name: Set environment variable ANDROID_HOME sdktools installation
  win_environment:
    state: present
    name: ANDROID_HOME
    value: C:\Users\winuser\AppData\Local\Android\Sdk
    level: machine

- name: Set PATH environment variable for all users
  win_path:
    elements:
      - '%ANDROID_HOME%'
      - '%JAVA_HOME%\bin'
    scope: machine

- name: Accept all licenses for sdk tools via sdk manager
  win_shell: 'for($i=0;$i -lt 30;$i++) { $response += "y`n"};
$response | .\sdkmanager.bat --licenses'
  args:
    chdir: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-
tools/latest/bin
  vars:
    ansible_become: yes
    ansible_become_method: runas
    ansible_become_user: admin
    ansible_become_password: "{{ vault_ansible_become_password
}}"

- name: Install platform via sdk manager
  win_command: 'sdkmanager.bat "emulator" "build-tools;29.0.3"
"platforms;android-29" "platform-tools" "patcher;v4" "sources;android-
29"'
  args:

```

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

```
chdir: C:/Users/winuser/AppData/Local/Android/Sdk/cmdline-  
tools/latest/bin  
vars:  
    ansible_become: yes  
    ansible_become_method: runas  
    ansible_become_user: admin  
    ansible_become_password: "{{ vault_ansible_become_password  
}}"
```

Παράρτημα Β΄

«Playbook» για την εισαγωγή Ηλεκτρονικού Υπολογιστή με λειτουργικό σύστημα «Windows» σε δίκτυο τομέα «Windows Domain Network»

```
---

- name: Join windows host to windows domain
  hosts: winhost

  tasks:
    - name: Join to Domain test.local
      win_domain_membership:
        dns_domain_name: test.local
        hostname: win10pro
        domain_admin_user: test.local\joindomain
        domain_admin_password: "{{ vault_domain_admin_password }}"
        state: domain
      register: domain_state
    - name: Reboot winhost
      win_reboot:
        vars:
          ansible_winrm_operation_timeout_sec: 200
          ansible_winrm_read_timeout_sec: 600
          ansible_winrm_transport: ntlm
          ansible_user: test.local\joindomain
          ansible_password: "{{ vault_domain_admin_password }}"
        when: domain_state.reboot_required
```

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

Παράρτημα Γ΄

«Playbook» για την εξαγωγή Ηλεκτρονικού Υπολογιστή με λειτουργικό σύστημα «Windows» από δίκτυο τομέα «Windows Domain Network»

```
---

- name: Unjoin windows host from windows domain
  hosts: winhost

  vars:
    ansible_user: test.local\joindomain
    ansible_password: "{{ vault_domain_admin_password }}"
    ansible_winrm_transport: ntlm

  tasks:
    - name: Unjoin from Domain test.local
      win_domain_membership:
        workgroup_name: WINWORKGROUP
        domain_admin_user: test.local\joindomain
        domain_admin_password: "{{ vault_domain_admin_password }}"
        state: workgroup

    - name: Reboot winhost
      win_reboot:
        vars:
          ansible_winrm_operation_timeout_sec: 200
          ansible_winrm_read_timeout_sec: 600
          ansible_winrm_transport: ntlm
          ansible_user: win10pro\admin
          ansible_password: "{{ vault_ansible_password }}"
```

Παράρτημα Δ΄

Εγκατάσταση λογισμικού προσομοίωσης «Simul8»

➤ 1^{ος} Τρόπος

```
---

- name: install Simul8 Software
  hosts: winhost
  vars:
    ansible_winrm_transport: ntlm
    ansible_user: test.local\joindomain
    ansible_password: "{{ vault_domain_admin_password }}"

  tasks:
    - name: Install Simul8
      win_shell: /s LIC:"1959-2368-3548" DIR:"C:\Program Files"
      args:
        chdir: \\testsrv\Shares\Simul8
```

➤ 2^{ος} Τρόπος

```
---

- name: install Simul8 Software
  hosts: winhost
  vars:
    ansible_winrm_transport: ntlm
    ansible_user: test.local\joindomain
    ansible_password: "{{ vault_domain_admin_password }}"

  tasks:
    - name: Install Simul8
      win_package:
```

“Αυτοματοποιημένη παραμετροποίηση ηλεκτρονικού υπολογιστή με τη χρήση Ansible σε περιβάλλον MS Windows”

Ελισάβετ Αραμπατζή

```
path: \\testsrv\Shares\Simul8\StudentS8SETUP.EXE  
product_id: simul8  
arguments: /s LIC:"1959-2368-3548" DIR:"C:\Program Files"
```