



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  

---

UNIVERSITY OF PIRAEUS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ  
ΠΜΣ "ΠΛΗΡΟΦΟΡΙΚΗ"

# **Ανάπτυξη Εφαρμογής για παραμετροποίηση δικτύου με το Django Framework**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΩΝ

**Ιάσωνας Σιμώτας, Μαρία Μπαμπανέλου**

Επιβλέπων: Πάνος Γκοτσιόπουλος  
Καθηγητής ΠΑΠΕΙ

Επιβλέπων: Δουληγέρης Χρήστος  
Καθηγητής ΠΑΠΕΙ

Αθήνα, Μήνας Έτος





ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
UNIVERSITY OF PIRAEUS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ  
ΠΜΣ "ΠΛΗΡΟΦΟΡΙΚΗ"  
ΤΟΜΕΑΣ

# **Ανάπτυξη Εφαρμογής για παραμετροποίηση δικτύου με το Django Framework**

ΔΙΠΛΩΜΑΤΙΚΗ  
ΤΩΝ

**Ιάσωνας Σιμωτας, Μαρία Μπαμπανέλου**

Επιβλέπων: Πάνος Γκοτσιόπουλος  
Καθηγητής ΠΑΠΕΙ  
Επιβλέπων: Δουληγέρης Χρήστος  
Καθηγητής ΠΑΠΕΙ

Εγκρίθηκε από την κάτωθι τριμελή επιτροπή την 1<sup>η</sup> Ιανουαρίου 2019.

---

Όνομα Επώνυμο  
Καθηγητής

---

Όνομα Επώνυμο  
Καθηγητής

---

Όνομα Επώνυμο  
Αναπληρωτής Καθηγητής

---

Ιάσοντας Σιμώτας, Μαρία Μπαμπανέλου  
Πτυχιούχοι Μεταπτυχιακού ΠΜΣ Πληροφορικής

Copyright © Όνομα Επώνυμο, 2023

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

*Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.*

*Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιά.*

# Περίληψη

Περίληψη διπλωματικής εργασίας.

Η παρούσα διπλωματική εργασία έχει σκοπό τη μελέτη και την ανάπτυξη λογισμικού για την παραμετροποίηση δικτυακών συσκευών με βασικό Framework το Django της Python. Η ιδέα για την υλοποίηση αυτής της εφαρμογής στάθηκε το υπόβαθρό μας στο software development ,στο χώρο του networking και με αφορμή μια εφαρμογή ενός μηχανικού της Cisco αποφασίσαμε να πάρουμε σαν βάση αυτήν και κάποιες ακόμα δημοσιεύσεις στο χώρο και να τις πάμε ένα βήμα παρακάτω . Σκοπός μας είναι να εισάγουμε τις έννοιες του microservices kubernetes και να μπορέσουμε να φτιάξουμε μια εφαρμογή η οποία περιλαμβάνει γνώσεις γύρω από διάφορους τομείς της Πληροφορικής. Εκτενής βιβλιογραφία θα παρουσιαστεί στο τέλος



# **Abstract**

Diploma thesis abstract.

Keywords: Keyword 1, Keyword 2





# Ευχαριστίες

Ευχαριστούμε πολύ όσους μας στήριξαν που ενώ παράλληλα με τη δουλειά μας μπορέσαμε να τελειώσουμε και τη διπλωματική εργασία μας. Παρόλο που η δουλειά μας μας βοήθησε σε πολλά επίπεδα στην κατανόηση πολλών εννοιών η φύση της ήταν τέτοια που δε μας έδινε χρόνο στο να κάτσουμε και να τελειώσουμε την εργασία αυτή.

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>13</b>
1.1	Απαιτήσεις και προδιαγραφές . . . . .	15
<b>2</b>	<b>State of the art</b>	<b>17</b>
2.1	Η επανάσταση στο web development . . . . .	17
2.2	Frameworks και γιατί χρησιμοποιούνται . . . . .	17
2.3	CI/CD pipeline . . . . .	19
2.4	Τι είναι ο kubernetes . . . . .	19
<b>3</b>	<b>Τεχνολογίες</b>	<b>21</b>
3.1	Εισαγωγή . . . . .	21
3.2	GNS3 . . . . .	22
3.3	Cisco IOS . . . . .	22
3.4	Εικονικοποίηση . . . . .	23
<b>4</b>	<b>Virtual Environment Set up</b>	<b>25</b>
4.1	GNS3 Installation . . . . .	25
4.2	Connection Establishment . . . . .	26
4.3	Σύνδεση με Django Server . . . . .	28

# Κατάλογος Σχημάτων

2.1	Γενική αρχιτεκτονική του Django . . . . .	18
3.1	General Network Topology . . . . .	22
3.2	Virtualization Γενική αρχιτεκτονική . . . . .	24
3.3	Virtualization Γενική αρχιτεκτονική . . . . .	24
4.1	GNS3 homepage . . . . .	26
4.2	Cisco ssh config . . . . .	26
4.3	Cisco dhcp config . . . . .	27
4.4	Local PC-GNS3VM-CISCO IOS Connection Architecture . . . . .	27
4.5	SSH traffic . . . . .	27
4.6	SSH traffic . . . . .	27



# Κεφάλαιο 1

## Εισαγωγή

Φαίνεται ότι ο χρόνος που σπαταλάνε οι μηχανικοί δικτύωσης όταν εισέρχονται σε εξοπλισμό δικτύωσης για να εισάγουν χειροκίνητα εντολές όπως για τη διαμόρφωση των συσκευών ή να εισέρχονται σε διακομιστές για τη χειροκίνητη ρύθμιση μία προς μία μια λίστα συσκευών/δικτύων(access lists) είναι πολύ μεγάλος, συνεπώς η εποχή που όλα αυτά γινόντουσαν χειροκίνητα φτάνει στο τέλος της. Όλο και περισσότεροι/περισσότερες εταιρείες προωθούν την αυτοματοποίηση καθώς βλέπουν ότι κάθε ώρα που επενδύεται στην αυτοματοποίηση μεταφράζεται σε πολλές ώρες εργασίας που εξοικονομούνται.

Η αυτοματοποίηση αυτών των εργασιών με κάποια καλοφτιαγμένη λογική προγραμματισμού επιτρέπει τη διαμόρφωση εκατοντάδων συσκευών μέσα σε λίγα λεπτά, απομακρύνει τη δυνατότητα των λανθασμένων ρυθμίσεων που προέρχονται από ανθρώπινο λάθος, επιτρέπει την καταγραφή των αλλαγών διαμόρφωσης και έχει το πλεονέκτημα ότι καθιστά τη διαμόρφωση τεχνικές λεπτομέρειες διαφανείς στον χρήστη που πρόκειται να ξεκινήσει τη διαδικασία αυτοματοποίησης. Για παράδειγμα, μια εταιρεία θα μπορούσε να αναθέσει υπερβολικά σε μια ομάδα λειτουργίας που δεν έχει τεχνικές γνώσεις δικτύωσης και απλά παρέχοντάς τους μια συγκεκριμένο σύνολο εισόδων θα μπορούσαν να διαμορφώσουν για Χ σημεία πρόσβασης σε Υ δίκτυα ένα συγκεκριμένο ΣΣΙΔ με τις επιθυμητές παραμέτρους. Οι δεδομένες εισοδοί θα μπορούσαν να εισαχθούν από αυτούς σε μια εφαρμογή ιστού και ο υποκείμενος προγραμματισμός κώδικας θα έκανε τα υπόλοιπα. Τελικά αυτό μεταφράζεται σε ένα πολύ γρήγορο και αξιόπιστο πλάνο κατά το οποίο η παραμετροποίηση και η διαμόρφωση ενός δικτύου δεν θα χρειάζεται να γίνει από τους μηχανικούς δικτύωσης χειροκίνητα.

Μπορούν να επενδύσουν συνεπώς αυτόν τον επιπλέον χρόνο σε άλλες εργασίες όπως ο σχεδιασμός και έτσι οι πιο χρονοβόρες διαδικασίες να αυτοματοποιηθούν. Αλλά η αυτοματοποίηση δεν είναι μόνο κάνει θαύματα όσον αφορά τη διαμόρφωση, είναι επίσης εξαιρετική για την παρακολούθηση της κατάστασης των δικτύων/συσκευών/θυρών, την απόκτηση πληροφοριών για την υγεία των ασύρματων δικτύων και κάθε άλλες πληροφορίες που μπορούν να λάβουν από της δυκτυακές συσκευές.

Είναι σημαντικό να ληφθεί υπόψη ότι οι επαναλαμβανόμενες καθημερινές/εβδομαδιαίες εργασίες που απαιτούν τη συλλογή πληροφοριών είναι εξαιρετικοί υποψήφιοι για αυτοματοποίηση. Με μια αυτοματοποίηση που αναζητά τα απαιτούμενα δεδομένα και κάνει κάποια επεξεργασία οι απαιτούμενες πληροφορίες μπορούν να ληφθούν γρήγορα και να παρουσιαστούν στους μηχανικό και τον/την απαλλάσσει από το να συνδέεται χειροκίνητα σε πολλές συσκευές, να ελέγχει ορισμένων γραμμών διαμόρφωσης, κ.λπ.

Η αυτοματοποίηση συσκευών χρησιμοποιείται εδώ και πολλά χρόνια για τη διαχείριση βλαβών ή την παρακολούθηση του επιπέδου υπηρεσιών, αλλά με τις αυξανόμενες επιχειρηματικές ανάγκες προκύπτουν νέες προκλήσεις και νέες ευκαιρίες. Μία από αυτές τις ευκαιρίες είναι η επένδυση των εταιρειών σε αυτοματισμούς δικτύων. Η ραγδαία ανάπτυξη των σύγχρονων δικτύων στις επιχειρήσεις μαζί με τις νέες τεχνολογίες, όπως το Διαδίκτυο των πραγμάτων (IOT) και το υπολογιστικό νέφος που βασίζονται επίσης στο δίκτυο, οδήγησαν στην ανάγκη ανάπτυξης της δικτυακής υποδομής με αποτέλεσμα την αύξηση του φόρτου εργασίας. απαιτήσεις για την παροχή, τη συντήρηση, την παρακολούθηση και τη διαχείριση από το δίκτυο.

Οι μέθοδοι που χρησιμοποιούσαν μέχρι σήμερα οι μηχανικοί δικτύων δεν ήταν μόνο χρονοβόρες αλλά και απαιτούνταν και γνώσεις σχετικά με ιδιόκτητα πρωτόκολλα και τεχνολογίες. Σε μια προσπάθεια να μειώσουν το κόστος και να δημιουργήσουν αποτελεσματικότητα οι μηχανικοί δικτύου ανέπτυξαν το Network automation ως τεχνικές αυτοματοποίησης για την αυτοματοποίηση επαναλαμβανόμενων καθημερινών εργασιών. Με την υποστήριξη σχεδόν όλων των μεγάλων εταιρειών δικτύωσης (όπως η Cisco) δημιουργήθηκε μια κοινότητα ανοιχτού κώδικα που είχε ως στόχο την υλοποίηση εφαρμογών αυτοματοποίησης κυρίως με τη χρήση τυποποιημένων διεπαφών (SSH, REST) και γενικών γλωσσών προγραμματισμού όπως η python. Με τη χρήση της Python και μιας συλλογής ενοτήτων και συναρτήσεων θα προσπαθήσουμε να φτιάξουμε μία εφαρμογή που συνδέει όλα τα παραπάνω.

Παράλληλα η επανάσταση που έφερε η εισαγωγή της λογικής των microservices στον κλαδο της Πληροφορικής μπορεί να καθιστήσει την εφαρμογή αυτή ακόμα πιο αξιόπιστη γιατί μπορεί να συμβάλει στο σχεδιασμό ενός συστήματος λογισμικού με μεγαλύτερη αξιοπιστία καθώς και να προσφέρει όλες εκείνες της θετικές προεκτάσεις χρήσης αυτών. Θα γίνει λοιπόν μια προσπάθεια εισαγωγής τεχνολογιών διαχείρισης και ανάπτυξης microservices όπως kubernetes και containers. Τα πλεονεκτήματα της χρήσης της αρχιτεκτονική Microservices είναι ότι προσφέρουν μεγαλύτερη ευελιξία μέσω της ανεξαρτησίας των υπηρεσιών, επιτρέποντας στους οργανισμούς να γίνουν πιο ευέλικτοι όσον αφορά τον τρόπο με τον οποίο προσφέρουν νέες επιχειρηματικές δυνατότητες ή ανταποκρίνονται στις μεταβαλλόμενες συνθήκες της αγοράς. Αναλυτική παρουσίαση αυτών θα γίνει σε επόμενο κεφάλαιο.

Σας συσκευές θα χρησιμοποιήσουμε αυτές της Cisco καθώς υπάρχουν ήδη βιβλιοθήκες οι οποίες υλοποιούν τα πρωτόκολλα επικοινωνίας και τις λειτουργίες που εμείς θέλουμε να υλοποιήσουμε. Η ανάπτυξη λογισμικού τέτοιων βιβλιοθηκών είναι

αντικείμενο μελέτης διπλωματικής εργασίας καθώς ξεφεύγει από τα πλαίσια μια μεταπτυχιακής διατριβής.

### **1.1 Απαιτήσεις και προδιαγραφές**

- GNS3 VM ,Cisco Images και GNS3 περιβάλλον
- Vs Code development environment
- Virtual box ή οποιονδήποτε type B hypervisor





## Κεφάλαιο 2

# State of the art

### 2.1 Η επανάσταση στο web development

Στην αρχή, οι εφαρμογές ιστού δεν ήταν τίποτα περισσότερο από ένα μάτσο HTML, CSS και javascript που ήταν τοποθετημένα μαζί, συνδεδεμένα μεταξύ τους. Ένας καλός προγραμματιστής ήταν σε θέση να φτιάξει σπουδαίες εφαρμογές ιστού αν αυτός/αυτή είχε αρκετές δεξιότητες/γνώσεις.

Στην εποχή μας, εμφανίστηκαν τα frameworks και λαμβάνοντας υπόψη ότι δεν βελτιώνουν αυτό που τελικά βλέπει ο χρήστης και ο/η αλληλεπιδράσεις του με το frontend που τελικά είναι ο τελικός στόχος, τότε θα μπορούσε κανείς να αναρωτηθεί γιατί χρησιμοποιούνται ευρέως στις μέρες μας. Παρόμοιες δουλειές υπάρχουν και σε άλλες διπλωματικές εργασίες καθώς και σε μη διπλωματικές εργασίες. Μηχανικοί από όλο τον κόσμο ασχολούνται με την αυτοματοποίηση συστημάτων και τη δημιουργία κώδικα που να αυτοματοποιεί συσκευές/συστήματα.

Με βάση άλλες τέτοιες προσπάθειες που έχουν γίνει στο παρελθόν εμείς συλλέξαμε την έως τώρα βιβλιογραφία και θα προσπαθήσουμε να φτιάξουμε μία τέτοια εφαρμογή η οποία όμως να βασίζεται στα τωρινά δεδομένα και να ενσωματώσουμε τις τελευταίες τεχνολογίες αιχμής όπως το Cloud Native. Θα γίνει προσπάθεια να δοθεί Εκτενής ανάλυση στο πως λειτουργεί η εφαρμογή καθώς και η αλληλεπίδραση της με τα συστήματα.

### 2.2 Frameworks και γιατί χρησιμοποιούνται

Στην τωρινή εποχή η ανάπτυξη λογισμικού είναι στενά συνδεδεμένη με τα frameworks. Η σύνδεση αυτή δεν είναι τυχαία καθώς η χρήση αυτών έχει κάνει τη ζωή των μηχανικών ανάπτυξης λογισμικού ευκολότερη. Θα εξηγήσουμε παρακάτω τους λόγους που συμβαίνει αυτό στο πλαίσιο κυρίως της δικιάς μας διπλωματικής εργασίας.

- **Modularity** Καθώς η εφαρμογή μεγαλώνει, ο κώδικας πρέπει να είναι καλά δομημένος σε φακέλους και αρχεία ανάλογα με το τι κάνει ο κώδικας. Στο πα-



Σχήμα 2.1: Γενική αρχιτεκτονική του Django

ρελθόν, οι μεγάλες εφαρμογές υπέφεραν όταν η εφαρμογή μεγάλωνε, υπήρχαν προβλήματα επεκτασιμότητας καθώς ο αριθμός των αρχείων javascript και CSS αυξανόταν ραγδαία και υπήρχαν πολύς επαναλαμβανόμενος κώδικας μεταξύ των αρχείων. Με την παροχή μιας καθορισμένης δομής, ένα συγκεκριμένο κομμάτι κώδικα μπορεί να αναζητηθεί εύκολα. Αν πάρουμε ως παράδειγμα παράδειγμα το πλαίσιο Django, το Django δομεί τον κώδικα σε ένα πολύ συγκεκριμένο τρόπο. Μέσα στο αρχείο `models.py` ορίζονται τα μοντέλα της βάσης δεδομένων. Αυτό συμβαίνει προκειμένου να μπορούν να γίνουν ερωτήματα στη βάση δεδομένων που δεν σχετίζονται με τη συγκεκριμένη βάση δεδομένων που χρησιμοποιείται στο έργο. Μέσα στο αρχείο `views.py` γίνεται η λογική για την ανάκτηση και την επεξεργασία των δεδομένων όταν το ζητάει ο χρήστης. Μέσα στο αρχείο `urls.py` υλοποιείται η δρομολόγηση της εφαρμογής. Μέσα στο φάκελο `templates` υπάρχουν όλα τα `.html` αρχεία στα οποία το αρχείο `views.py` στέλνει τα δεδομένα που λαμβάνει για να τα απεικονίσει, κ.λπ.

- **Ταχύτερη ανάπτυξη** Τα πλαίσια παρέχουν έτοιμες λειτουργίες, καλώντας απλώς τις ήδη ενσωματωμένες συναρτήσεις/μεθόδους. Διαβάζοντας απλώς την τεκμηρίωση του πλαισίου και μαθαίνοντας πώς να τις χρησιμοποιεί, ο προγραμματιστής μπορεί να ενσωματώσει λειτουργικότητες που διαφορετικά θα ήταν δύσκολο να υλοποιήσει και επίσης πολύ χρονοβόρες. Παραδείγματα περιλαμβάνουν λειτουργίες ελέγχου ταυτότητας, λειτουργίες διαχείρισης συνεδριών, λειτουργίες λειτουργίας βάσεων δεδομένων, λειτουργίες επικύρωσης φορμών και λειτουργίες για την παροχή ασφάλειας έναντι κακόβουλων επιτιθέμενων.
- **Επέκταση κώδικα** Τα περισσότερα frameworks επιτρέπουν την επέκταση κάποιου κομματιού κώδικα που θα χρησιμοποιηθεί σε πολλά άλλα αρχεία. Αυτό εξασφαλίζει ότι δεν υπάρχει επαναλαμβανόμενος κώδικας και οποιαδήποτε αλλαγή σε αυτόν τον κώδικα μεταφράζεται σε όλες τις περιπτώσεις που χρησιμοποιούν αυτόν τον κώδικα.
- **Ευκολότερη αναγνωσιμότητα του κώδικα** Δεδομένου ότι ο κώδικας χρησιμοποιεί καλά καθορισμένες τυποποιημένες συναρτήσεις και μια συγκεκριμένη δομή, είναι ευκολότερο να τον καταλάβει κάποιος που είναι νέος στο κώδικα αλλά γνωρίζει πώς λειτουργεί το πλαίσιο

## 2.3 CI/CD pipeline

Μόλις η εφαρμογή ιστού συνδεθεί με το απομακρυσμένο αποθετήριο, η τελευταία τάση στο στον κόσμο του DevOps είναι η υλοποίηση ενός αγωγού CI/CD, ο οποίος ουσιαστικά είναι μια αυτοματοποιημένη διαδικασία που ενεργοποιείται όταν νέος κώδικας δημοσιεύεται στο απομακρυσμένο αποθετήριο. Αυτή η διαδικασία ξεκινάει τη δημιουργία κώδικα, εκτελεί κάποιες δοκιμές και τέλος, αν όλα είναι εντάξει, αναπτύσσει αυτόματα τον κώδικα στην παραγωγή περιβάλλον. Με αυτόν τον τρόπο, οι προγραμματιστές μπορούν να διασφαλίσουν ότι τίποτα δεν θα χαλάσει στην παραγωγή και οι νέες λειτουργικότητες εξυπηρετούνται το συντομότερο δυνατό στους πελάτη. Στην περίπτωσης μας τόσο η διπλωματική εργασία(latex) όσο και η εφαρμογή υλοποιήθηκαν με αυτή τη λογική.

## 2.4 Τι είναι ο kubernetes

Ο κυβερνήτης είναι ο διαχειριστής των με απλά λόγια ο διαχειριστής των containers. Είναι μια πλατφόρμα ανοικτού κώδικα για τη διαχείριση φορτίων εργασίας και υπηρεσιών που περιέχουν containers , η οποία διευκολύνει τόσο τη δηλωτική διαμόρφωση όσο και την αυτοματοποίηση. Διαθέτει ένα μεγάλο, ταχέως αναπτυσσόμενο οικοσύστημα. Οι υπηρεσίες, η υποστήριξη και τα εργαλεία του Kubernetes είναι ευρέως διαθέσιμα.



## Κεφάλαιο 3

# Τεχνολογίες

### 3.1 Εισαγωγή

Αρχικά δημιουργήθηκε η βασική δομή και η δομή της διαδικτυακής πύλης καθορίστηκε. Αυτή η βασική δομή φαίνεται στο Παρακάτω σχήμα

Είναι σημαντικό να σημειωθεί ότι ολόκληρη η εφαρμογή Δθανγο βρίσκεται στον ίδιο φυσικό διακομιστή. Όταν ο χρήστης στέλνει ένα αίτημα για την εκτέλεση ενός σεναρίου με συγκεκριμένες εισόδους αυτό στέλνεται στις συσκευές στο τοπικό δίκτυο.

Το πρώτο βήμα στη διαδικασία ήταν να καθοριστεί τι θα αυτοματοποιηθεί με βάση τις διάφορες εκτιμήσεις. Για να καθοριστεί αυτό, πραγματοποιήθηκαν πολλές συναντήσεις καταιγισμού ιδεών. με την ομάδα. Προτού γίνει αυτό όμως η αρχική ιδέα που τέθηκε στο τραπέζι βγήκε με βάση μια παρόμοια δουλειά ενός μηχανικού της Cisco. Το έργο του θα αναφερθεί αναλυτικά στην εκτενή βιβλιογραφία στο τέλος. Με βάση λοιπόν αυτο το έργο ξεκινήσαν συζητήσεις για το πως θα μπορέσουμε να αναπτύξουμε κάτι παρόμοιο καθώς και να το εμπλουτίσουμε στο τέλος έτσι ώστε να ανταπρίνεται όσο γίνεται στις τεχνολογίες του σήμερα.

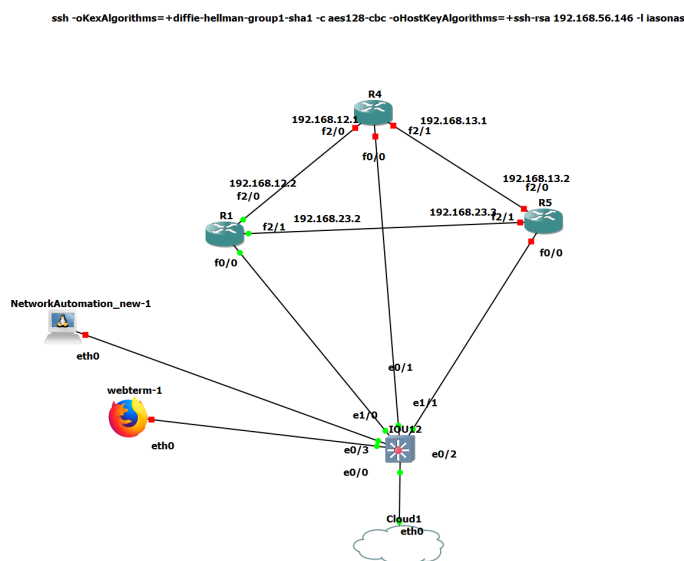
Σε αυτές τις συναντήσεις που γίνανε μεταξύ μας τέθηκαν πολλές ιδέες τέθηκαν στο τραπέζι και η ομάδα καθόρισε μια σειρά προτεραιότητας για την ανάπτυξη. Ο στόχος σε πολλούς αυτοματισμούς είναι να μειωθεί ο χρόνος που καταναλώνεται για την εκτέλεση αυτές τις επαναλαμβανόμενες εργασίες. Πολλές τεχνολογίες χρησιμοποιήθηκαν για την υλοποίηση του συγκεκριμένου έργου οι οποίες θα παρουσιαστούν εκτενώς σε άλλες ενότητες.

Η υλοποίησή μιας τέτοιας εφαρμογής είχε κάποιες δυσκολίες. Κυρίως ποιο θα είναι το περιβάλλον στο οποίο η εφαρμογή θα μπορούσε να τεσταριστεί και υλοποιηθεί. Για αυτό πάρθηκε η απόφαση οι συσκευές με τις οποίες θα τεσταριστεί και συνάμα θα λειτουργήσει η εφαρμογή θα είναι virtual συσκευές της Cisco οι οποίες θα τρέχουν στο GNS3 και το GNS3 θα μπορεί να επικοινωνεί δικτυακά με τον Django Server στο τοπικό δίκτυο. Το στήσιμο όλου του περιβάλλοντος και της εφαρμογής θα αναλυθεί εκτενώς περαιτέρω σε άλλο κεφάλαιο.

## 3.2 GNS3

Το GNS3 Είναι ένα εργαλείο προσομοίωσης δικτύων ανοικτού κώδικα που επιτρέπει στους χρήστες να προσομοιάσουν σύνθετες τοπολογίες δικτύων στους υπολογιστές τους. Μηχανικοί δικτύων και φοιτητές το χρησιμοποιούν ευρέως για να μάθουν και να εξασκηθούν σε έννοιες δικτύωσης, να δοκιμάσουν διαμορφώσεις δικτύου και να δημιουργήσουν εικονικά περιβάλλοντα δικτύου.

Το GNS3 υποστηρίζει διάφορες συσκευές δικτύου, όπως δρομολογητές, μεταγωγείς και τείχη προστασίας από διάφορους προμηθευτές, συμπεριλαμβανομένων των Cisco, Juniper, Nokia και άλλων. Επιτρέπει στους χρήστες να προσομοιάσουν διάφορα σενάρια και διαμορφώσεις δικτύου και να δοκιμάσουν τη συμπεριφορά των συσκευών δικτύου σε ένα ελεγχόμενο περιβάλλον.



Σχήμα 3.1: General Network Topology

## 3.3 Cisco IOS

Το IOU σημαίνει IOS on Unix είναι μια εικονική έκδοση του λογισμικού IOS της Cisco που μπορεί να χρησιμοποιηθεί για σκοπούς προσομοίωσης και δοκιμής δικτύου. Επιτρέπει στους μηχανικούς δικτύου να δημιουργούν εικονικές τοπολογίες δικτύου και να εξασκούνται σε διάφορες εργασίες δικτύου, όπως η διαμόρφωση δρομολογητών και μεταγωγέων, χωρίς να απαιτείται φυσικό υλικό. Το πλεονέκτημα του GNS3 σε σχέση με εφαρμογές άλλες όπως το Packet tracer είναι ότι το GNS3 μπορεί να σηκώσει πραγματικά images άρα πραγματικό λογισμικό συνεπώς οι λειτουργίες που μπορείς να κάνεις είναι πολύ περισσότερες.

Το Cisco IOU χρησιμοποιείται συχνά σε συνδυασμό με λογισμικό προσομοίωσης δικτύου όπως GNS3 ή EVE-NG, τα οποία αποτελούν την πλατφόρμα εικονικοποίησης δικτύου που σας επιτρέπει να δημιουργείτε και να διαχειρίζεστε εικονικά

περιβάλλοντα δικτύου για σκοπούς δοκιμής και εκμάθησης, τα οποία παρέχουν ένα γραφικό περιβάλλον χρήστη για τη δημιουργία και τη διαχείριση εικονικών τοπολογιών δικτύου. Οι εικόνες IOU μπορούν να φορτωθούν σε αυτά τα εργαλεία προσομοίωσης για τη δημιουργία εικονικών συσκευών Cisco που μπορούν να διαμορφωθούν και να δοκιμαστούν όπως το φυσικό δίκτυο συσκευές.

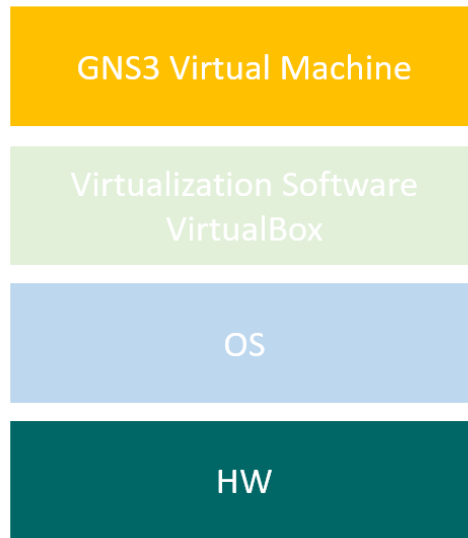
### **3.4 Εικονικοποίηση**

Στην επιστήμη της πληροφορικής, η εικονικοποίηση virtualization είναι ένας ευρύς όρος των υπολογιστικών συστημάτων που αναφέρεται σε έναν μηχανισμό αφαίρεσης, στοχευμένο στην απόκρυψη λεπτομερειών της υλοποίησης και της κατάστασης ορισμένων υπολογιστικών πόρων από πελάτες των πόρων αυτών (π.χ. εφαρμογές, άλλα συστήματα, χρήστες κλπ). Η εν λόγω αφαίρεση μπορεί είτε να αναγκάζει έναν πόρο να συμπεριφέρεται ως πλειάδα πόρων (π.χ. μία συσκευή αποθήκευσης σε διακομιστή τοπικού δικτύου), είτε πολλαπλούς πόρους να συμπεριφέρονται ως ένας (π.χ. συσκευές αποθήκευσης σε καταναμημένα συστήματα).

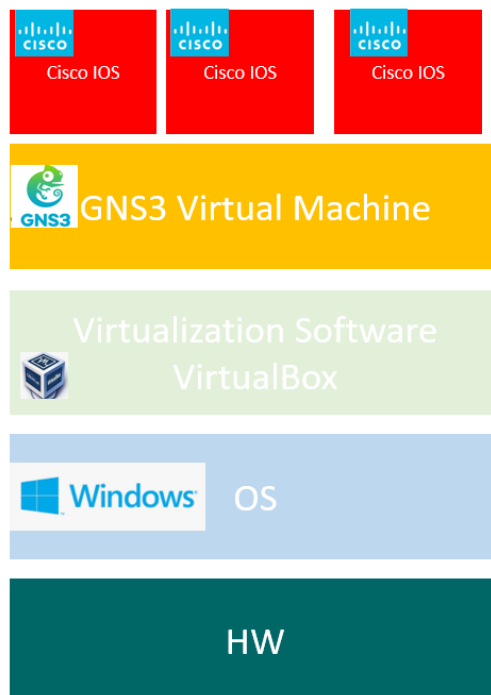
Η εικονικοποίηση δημιουργεί μία εξωτερική διασύνδεση η οποία αποκρύπτει την υποκείμενη υλοποίηση (π.χ. πολυπλέκοντας την πρόσβαση από διαφορετικούς χρήστες). Αυτή η προσέγγιση στην εικονικοποίηση αναφέρεται ως εικονικοποίηση πόρων. Μία άλλη προσέγγιση, ίδιας όμως νοοτροπίας, είναι η εικονικοποίηση πλατφόρμας, όπου η αφαίρεση που επιτελείται προσομοιώνει ολόκληρους υπολογιστές. Το αντίθετο της εικονικοποίησης είναι η διαφάνεια: ένας εικονικός πόρος είναι ορατός, αντιληπτός, αλλά στην πραγματικότητα ανύπαρκτος, ενώ ένας διαφανής πόρος είναι υπαρκτός αλλά αόρατος.

Θα εξηγήσουμε την εικονικοποίηση στην δικιά μας περίπτωση. Το πρώτο επίπεδο είναι αυτό του υλικού. Η εικονικοποίηση σε τεχνολογία εικονοποιεί το υλικό για να μπορέσει να δώσει πόρους στις εικονικές μηχανές. Η υλοποίηση της εικονικοποίησης γίνεται με λογισμικό hypervisor. Στη δικιά μας περίπτωση ο hypervisor είναι το Virtual Box ο οποίος είναι ένας τύπου B hypervisor. Ο hypervisor τύπου 2 είναι μια εφαρμογή εγκατεστημένη στο λειτουργικό σύστημα του κεντρικού υπολογιστή το οποίο μας δίνει τη δυνατότητα να σηκώσουμε εικονικές μηχανές άλλων λειτουργικών συστημάτων πάνω στο ήδη υπάρχον σύστημα.

Οι παρακάτω εικόνες μπορούν να εξηγήσουν σχηματικά τη γενική καθώς και την ειδική αρχιτεκτονική.



Σχήμα 3.2: Virtualization Γενική αρχιτεκτονική



Σχήμα 3.3: Virtualization Γενική αρχιτεκτονική



## Κεφάλαιο 4

# Virtual Environment Set up

### 4.1 GNS3 Installation

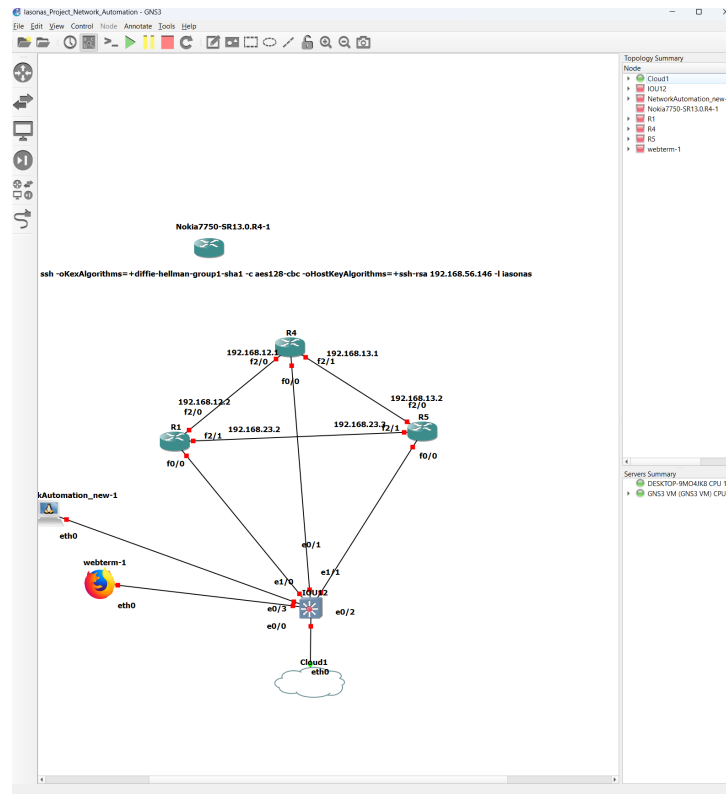
Το GNS3 είναι ένα λογισμικό που χρησιμοποιείται για την εξομίωση, τη διαμόρφωση και τη δοκιμή ενός περιβάλλοντος δικτύου. Είναι ένα ελεύθερο λογισμικό ανοικτού κώδικα και μπορείτε να το κατεβάσετε από τον επίσημο δικτυακό τόπο <https://www.gns3.com/>. Το GNS3 αποτελείται από δύο στοιχεία. Το ολοκληρωμένο λογισμικό (GUI) το οποίο είναι ένα γραφικό διεπαφή χρήστη και την εικονική μηχανή (VM), η οποία είναι ένας διακομιστής που εκτελείται σε εικονικό περιβάλλον και παρέχει καλύτερο μέγεθος τοπολογίας και υποστήριξη συσκευών. Η εγκατάσταση είναι απλή και θα πρέπει να χρησιμοποιούνται οι προεπιλεγμένες επιλογές.

Για να γίνει σωστά η εγκατάσταση θα πρέπει το software version του GNS3 να είναι το ίδιο με το software version του GNS3 VM. Όταν λοιπόν γίνει η εγκατάσταση και ανοίγουμε το GNS3 GUI αυτή η ενέργεια θα κάνει trigger το booting του GNS3 VM. Μόλις γίνει η εγκατάσταση μπορεί να ανοίξει η εφαρμογή και να κάνουμε import cisco IOS images. Στην παρακάτω εικόνα μπορούμε να δούμε τι γίνεται όταν ανοίγουμε το GNS3.

Προκειμένου να μπορέσει να επικοινωνήσει το PC μας στο τοπικό δίκτυο με το GNS3 VM στο τοπικό δίκτυο θα πρέπει να γίνουν κάποιες ρυθμίσεις τόσο στο GNS3 VM όσο και στις συσκευές της Cisco.

Στις συσκευές της Cisco θα πρέπει να γίνει η παρακάτω παραμετροποίηση όπως εμφανίζεται στις εικόνες 4.1,4.2,4.3.

Μέχρι αυτή τη στιγμή έχουμε παραμετροποιήσει τις συσκευές με τέτοιο τρόπο ώστε να δέχονται απομακρυσμένη σύνδεση. Τώρα θα εξηγήσουμε πως μπορούμε να φτιάξουμε την επικοινωνία μεταξύ εικονικών μηχανών της Cisco και του τοπικού μας υπολογιστή. Η λογική είναι ότι η συσκευή Cloud θα μας επιτρέψει να φτιάξουμε τη σύνδεση αυτή. Η εικόνα 4.4 μας παρουσιάζει σε ανώτερο επίπεδο τη λογική αυτή σύνδεση.



Σχήμα 4.1: GNS3 homepage

## 4.2 Connection Establishment

Όταν λοιπόν γίνει αυτή η παραμετροποίηση και τοπολογία θα πρέπει όλα αυτά τα διαφορετικά ζευγάρια να ανήκουν στο ίδιο τοπικό δίκτυο. Η εικονική διεπαφή από την οποία θα περνάει όλη η κίνηση είτε μιλάμε για REST είτε για SSH θα είναι η eth0 στο GNS3 VM. Παρακάτω ένα trace στην εικόνα 4.5 που συλλέχθηκε αποδεικνύει ότι η σύνδεση πραγματοποιείται χωρίς προβλήματα.

Προκειμένου να γίνει η συλλογή του συγκεκριμένου trace χρησιμοποιήθηκε η παρακάτω εντολή: `tcpdump -i eth0 -w /home/gns3/test.pcap`. Η συλλογή του trace έγινε με το πρωτόκολλο SFTP.

```

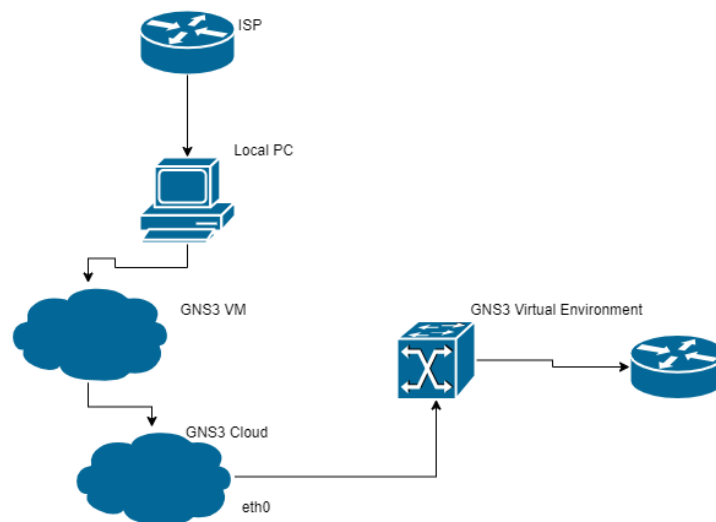
line vty 0 4
login local
transport input ssh
line vty 5 15
login local
transport input ssh

```

Σχήμα 4.2: Cisco ssh config

```
interface FastEthernet0/0
ip address dhcp
duplex half
,
```

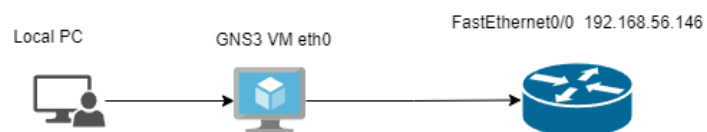
Σχήμα 4.3: Cisco dhcp config



Σχήμα 4.4: Local PC-GNS3VM-CISCO IOS Connection Architecture

226 16.412596	192.168.56.1	192.168.56.146	SSHv2	106 Client: Encrypted packet (len=52)
227 16.419894	192.168.56.146	192.168.56.1	SSHv2	106 Server: Encrypted packet (len=52)
228 16.420665	192.168.56.1	192.168.56.146	TCP	54 50708 → 22 [ACK] Seq=2758 Ack=1764 Win=63877 Len=0
229 16.532536	192.168.56.1	192.168.56.146	SSHv2	106 Client: Encrypted packet (len=52)
230 16.538112	192.168.56.146	192.168.56.1	SSHv2	106 Server: Encrypted packet (len=52)
231 16.538564	192.168.56.1	192.168.56.146	TCP	54 50708 → 22 [ACK] Seq=2810 Ack=1816 Win=63877 Len=0
232 16.657343	192.168.56.146	192.168.56.1	SSHv2	122 Server: Encrypted packet (len=68)
233 16.657392	192.168.56.146	192.168.56.1	SSHv2	98 Server: Encrypted packet (len=36)
234 16.657394	192.168.56.146	192.168.56.1	SSHv2	98 Server: Encrypted packet (len=36)

Σχήμα 4.5: SSH traffic



Σχήμα 4.6: SSH traffic

### **4.3 Σύνδεση με Django Server**