



Πανεπιστήμιο Δυτικής Αττικής

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών
Department of Informatics and Computer
Engineering

Εφαρμογή Τεχνολογίας SDN για Data Centers

Implementing SDN Technology in Data Centers

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Συγγραφέας:

ΝΙΚΟΥΛΗΣ ΒΑΣΙΛΕΙΟΣ (cs131014)

Επιβλέπων: Αντώνιος Μπόγρης

Καθηγητής Πα. Δ. Α.

Οκτώβριος 2020

Εξεταστική επιτροπή:

Καρκαζής Παναγιώτης

Επίκουρος Καθηγητής

Πανεπιστημίου Δυτικής Αττικής

Μάμαλης Βασίλειος

Καθηγητής

Πανεπιστημίου Δυτικής Αττικής

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η Νικούλης Βασίλειος του Κυριάκου, με αριθμό μητρώου cs131014 φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο/Η Δηλών/ούσα

Νικούλης Βασίλειος



Περίληψη

Το μέγεθος των Data Center μεγαλώνει εκθετικά, έτσι ώστε να ικανοποιήσει τις ανάγκες των εφαρμογών και των χρηστών. Το γεγονός αυτό έχει δημιουργήσει την ανάγκη να βρεθεί ένας τρόπος ελέγχου της ροής των δεδομένων μέσα σε αυτό.

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής τεχνολογίας SDN που ελέγχει τον ελεγκτή ενός SDN δικτύου με σκοπό τον έλεγχο της δρομολόγησης στο δίκτυο προκειμένου αυτή να επιτευχθεί με τον βέλτιστο τρόπο. Για την πτυχιακή επιλέχθηκε ο OpenDaylight controller για ελεγκτής ενώ για την αναπαράσταση της τοπολογίας επιλέχθηκε το Mininet που είναι εξομοιωτής δικτύων. Για την εκμάθηση της τοπολογίας του δικτύου καθώς και για την παραμετροποίησή του χρησιμοποιήθηκε το REST api του OpenDaylight.

Ευχαριστίες

Η παρούσα πτυχιακή εργασία εκπονήθηκε στο Πανεπιστήμιο Δυτικής Αττικής και σηματοδοτεί την ολοκλήρωση των σπουδών μου σε αυτό γι' αυτό θα ήθελα να εκφράσω τις ευχαριστίες μου στους ανθρώπους που στάθηκαν δίπλα μου όλο αυτόν τον καιρό .

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή κ. Αντώνιο Μπόγρη για την εμπιστοσύνη που μου έδειξε αναθέτοντάς μου την εκπόνηση της, καθώς και τη καθοδήγηση που μου παρείχε στην επιλογή αυτού του ενδιαφέροντος θέματος.

Παράλληλα, θα ήθελα να ευχαριστήσω τους φίλους μου που στάθηκαν δίπλα μου όλα αυτά τα χρόνια. Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου για την απόλυτη στήριξη τους κατά τη διάρκεια αυτού του εγχειρήματος.

Νικούλης Βασίλειος,
Αθήνα, Οκτώβριος 2020

Περιεχόμενα

Πανεπιστήμιο Δυτικής Αττικής.....	1
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών	1
1. Εισαγωγή.....	8
1.1 Τα Κέντρα Δεδομένων.....	8
1.2 Σημερινή Αρχιτεκτονική Δικτύων.....	8
1.3 Ανάγκη χρήσης καινούριας αρχιτεκτονικής.....	10
1.4 Σκοπός Διπλωματικής.....	10
2. Software Defined Networks	12
2.1 Δίκτυα Καθορισμένα από Λογισμικό	12
2.2 Εισαγωγή στο Software Defined Networking	12
2.3 Το πρωτόκολλο OpenFlow	14
2.4 Τα δομικά μέρη του OpenFlow	15
2.4.1 OpenFlow Controller	16
2.4.2 OpenFlow μεταγωγέας (OpenFlow Switch)	16
2.4.3 Πόρτες (Ports).....	17
2.4.4 OpenFlow Πίνακες (OpenFlow tables)	18
2.4.5 Ασφαλές Κανάλι (OpenFlow Channel)	19
3. Open Daylight Controller.....	20
3.1 Αρχιτεκτονική του OpenDaylight	20
3.2 Πλατφόρμα ελεγκτή.....	21
3.2.1 Base Network Service Functions	21
3.3 Rest Αρχιτεκτονική	22
3.4 Εγκατάσταση OpenDaylight	23
3.5 Mininet	24
3.6 Χαρακτηριστικά Mininet	26
3.7 Περιορισμοί του Mininet	27
3.8 Εγκατάσταση Mininet.....	27
3.9 Αναβάθμιση παλαιότερης έκδοσης Mininet	28
3.10 Έναρξη τοπολογίας.....	29
4. Data Centers	30
4.1 Εξέλιξη των Κέντρων δεδομένων στο πέρασμα των ετών.....	30
4.2 Απαιτήσεις Κέντρων Δεδομένων.....	33
4.3 Προκλήσεις στα κέντρα δεδομένων.....	35

5. OpenStack.....	37
5.1 Τι είναι το OpenStack	37
5.2 Αρχιτεκτονική του OpenStack	38
5.2.1 Keystone	41
5.2.2 Glance.....	42
5.2.3 Nova.....	43
5.2.4 OpenStack Dashboard	44
5.3 OpenStack Dashboard (Προεπισκόπηση)	45
Υλοποίηση και συμπεράσματα	53
Μελλοντικές αναβαθμίσεις.....	56
Βιβλιογραφία	58

1. Εισαγωγή

1.1 Τα Κέντρα Δεδομένων

Τα κέντρα δεδομένων είναι φυσική ή εικονική υποδομή που χρησιμοποιείται από επιχειρήσεις για στέγαση των υπολογιστών, των εξυπηρετητών, των δικτυακών συστημάτων καθώς και των απαιτούμενων στοιχείων από το τμήμα IT (*Information Technology*). Οι ανάγκες που εξυπηρετεί συνήθως συνοψίζονται σε αποθήκευση και επεξεργασία καθώς και εξυπηρέτηση μεγάλων ποσοτήτων, κριτικής σημασίας δεδομένων για τους πελάτες σε αρχιτεκτονικές πελάτη-εξυπηρετητή (*client-server architectures*).

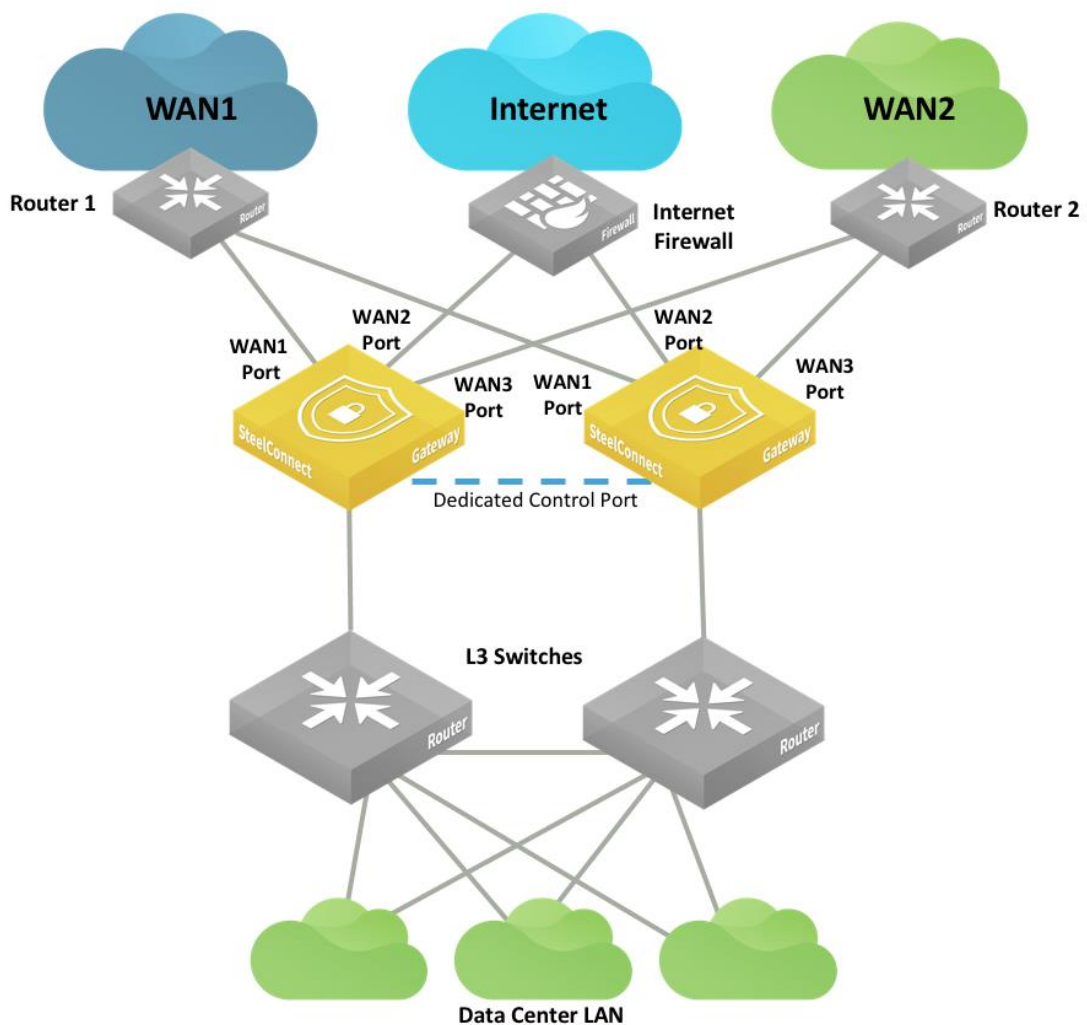
Ένα τέτοιο δίκτυο απαιτεί εκτεταμένα και εφεδρικά συστήματα παροχής ενέργειας, συστήματα ψύξης, υψηλού επιπέδου, ασφάλεια και πλεονάζουσες δικτυακές συνδέσεις.

Η διαχείριση των κέντρων δεδομένων περιλαμβάνει την εξασφάλιση της αξιοπιστίας στις συνδέσεις που αυτό περιλαμβάνει αλλά και την φύλαξη των αποθηκευμένων σε αυτό πληροφοριών. Ιδιαίτερα σημαντική παράμετρος είναι, όμως, και η διαχείριση του φορτίου κατά τον πιο οικονομικό τρόπο.

1.2 Σημερινή Αρχιτεκτονική Δικτύων

Τα δίκτυα υπολογιστών διαδραματίζουν ένα κρίσιμο ρόλο στη σημερινή κοινωνία. Οι υπηρεσίες που προσφέρει το διαδίκτυο όπως είναι για παράδειγμα, οι μηχανές αναζήτησης, τα μέσα κοινωνικής δικτύωσης καθώς και υπηρεσίες ηλεκτρονικού εμπορίου φιλοξενούνται σε κέντρα δεδομένων. Εκεί εκατοντάδες υπολογιστές συνδέονται με μεγάλα δίκτυα δεδομένων. Τα κέντρα δεδομένων διασυνδέονται μεταξύ τους με δίκτυα ευρείας περιοχής (WAN) που καλύπτουν ολόκληρο τον πλανήτη. Οι τελικοί χρήστες χρησιμοποιώντας τους προσωπικούς τους υπολογιστές, τα κινητά τους τηλέφωνα αλλά και άλλες συσκευές επιζητούν πρόσβαση στις υπηρεσίες που προσφέρει το διαδίκτυο μέσω διαφορετικών τεχνολογιών.

Τα τωρινά δίκτυα υπολογιστών αποτελούνται από συσκευές όπως οι δρομολογητές (routers), οι μεταγωγείς (switches), τα τείχος προστασίας (firewall), οι διακομιστές (servers) και ο λόγος χρήσης τους είναι η πραγματοποίηση συγκεκριμένων λειτουργιών σε ένα δίκτυο. Για παράδειγμα, ο router είναι υπεύθυνος για την δρομολόγηση της κίνησης μεταξύ δύο τοπικών δικτύων (LANs), το switch είναι η συσκευή που χρησιμοποιείται κατά κόρον πλέον για την σύνδεση των τερματικών ενός LAN και οι υπηρεσίες που προσφέρει είναι η προώθηση της κίνησης εντός του δικτύου. Επίσης, τα firewalls (χρησιμοποιούνται είτε σαν συσκευές, είτε υλοποιούνται μέσω software) ορίζουν το είδος της κίνησης που επιτρέπεται να περάσει σε ένα δίκτυο. Τέλος, οι servers (είτε πρόκειται για φυσική συσκευή υλικού, είτε για εικονικές μηχανές που λειτουργούν πάνω από ένα Server) είναι το μέρος όπου βρίσκεται μία ποικιλία εφαρμογών και υπηρεσιών και στις οποίες οι τελικοί χρήστες και οι δικτυακές συσκευές επιθυμούν να έχουν πρόσβαση.



1.3 Ανάγκη χρήσης καινούριας αρχιτεκτονικής

Η αύξηση των κινητών συσκευών σε συνδυασμό με τις εφαρμογές που τις συνοδεύουν, η εικονοικοποίηση (Virtualization) των servers και η άφιξη των cloud υπηρεσιών είναι οι λόγοι που οδηγούν την δικτυακή βιομηχανία να επανεξετάσει την παραδοσιακή δικτυακή αρχιτεκτονική. Αυτή η σχεδίαση είχε νόημα όταν η υπολογιστική client-server επικρατούσε, αλλά μία τέτοια στατική αρχιτεκτονική είναι απρόσφορη στην δυναμική υπολογιστική και στις ανάγκες αποθήκευσης των σημερινών κέντρων δεδομένων. Κάποιες από τις τάσεις της υπολογιστικής που απαιτούν την αναδιαμόρφωση της τρέχουσας αρχιτεκτονικής είναι:

- Αλλαγή στον τρόπο κίνησης των δεδομένων
- Η εμπορευματοποίηση του IT
- Η άνοδος των cloud υπηρεσιών
- Η ανάγκη για μεγαλύτερο bandwidth λόγω χειρισμού πολλών δεδομένων

1.4 Σκοπός Διπλωματικής

Σε αυτήν τη διπλωματική εργασία γίνεται συλλογή στατιστικών στοιχείων σε Ευφυή Προγραμματιζόμενα Δίκτυα (Software Defined Networks) μικρής ή και μεγάλης κλίμακας προκειμένου να χρησιμοποιηθούν για την εύρεση αποδοτικότερης δρομολόγησης εντός του δικτύου σε πραγματικό χρόνο.

Στα πλαίσια της εργασίας χρησιμοποιήθηκε ο OpenDaylight Controller (ODL) μέσω του οποίου ο διαχειριστής μπορεί απομακρυσμένα και από οποιοδήποτε μηχάνημα εκτός του δικτύου να συλλέξει πληροφορίες για το δίκτυο, αλλά και να του στείλει επίσης πίσω κατάλληλες παραμετροποιήσεις για την διαφοροποίηση της λειτουργίας του. Η τοπολογία τρέχει στο εργαλείο εξομίωσης δικτύων Mininet.

Αρχικά συλλέγονται πληροφορίες για την τοπολογία του εκάστοτε δικτύου μέσω του OpenDaylight Controller. Ανάλογα με τις πληροφορίες αυτές στέλνονται κατάλληλες παραμετροποιήσεις για την τροποποίηση στοιχείων και στη συνέχεια γίνεται συλλογή, επεξεργασία και εξαγωγή των στατιστικών. Η εφαρμογή τρέχει

διαρκώς σε πραγματικό χρόνο, προκειμένου να ελέγχονται συνεχώς όλα τα δεδομένα για τυχόν αλλαγές που χρειαστεί να γίνουν έτσι ώστε να επιτευχθεί η βέλτιστη λύση.

2. Software Defined Networks

2.1 Δίκτυα Καθορισμένα από Λογισμικό

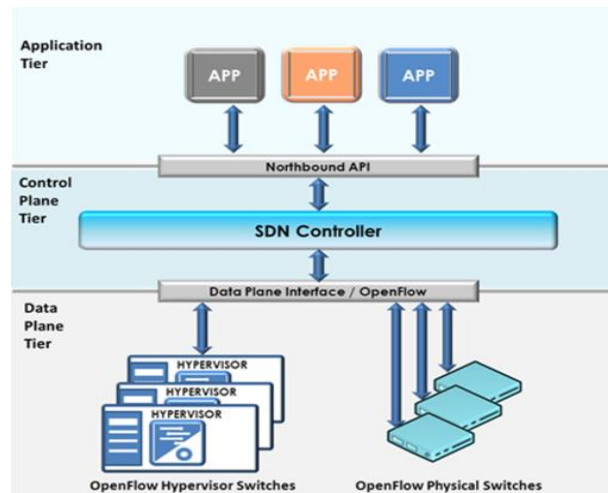
Η εξάπλωση των κινητών ευφυών συσκευών με τις εφαρμογές που τις συνοδεύουν, η εικονοικοποίηση (Virtualization) των Servers καθώς και η άφιξη των Cloud υπηρεσιών οδηγούν στη δημιουργία μιας νέας αρχιτεκτονικής δικτύων προκειμένου να ικανοποιηθούν οι νέες ανάγκες στις οποίες τα παραδοσιακά δίκτυα αδυνατούσαν να ανταπεξέλθουν. Έτσι με τη βοήθεια του Ανοικτού Λογισμικού (Open Source), εισήχθη στη ζωή μας η νέα γενιά αρχιτεκτονικής δικτύων, τα Δίκτυα Καθορισμένα από Λογισμικό (Software Defined Networks).

Το Software Defined Networking (SDN) δίνει την δυνατότητα στους διαχειριστές του δικτύου (Network Admin) να μπορούν άμεσα να ελέγξουν το δίκτυο. Το SDN χρησιμοποιεί τις ίδιες συσκευές όπως τα παραδοσιακά δίκτυα αλλά τα κάνει πιο ευέλικτα διαχωρίζοντας το επίπεδο Ελέγχου από το επίπεδο Δεδομένων.

Το επίπεδο Ελέγχου είναι υπεύθυνο για τον καθορισμό των διαδρομών και των λογικών διασυνδέσεων των κόμβων ενώ το επίπεδο Δεδομένων για την μεταφορά τους. Στόχος, λοιπόν, της νέας αυτής αρχιτεκτονικής είναι να μετριάσει περιορισμούς όπως πολυπλοκότητα, ασυνέπειες στις δικτυακές πολιτικές που εφαρμόζονται ή θέματα επεκτασιμότητας.[1]

2.2 Εισαγωγή στο Software Defined Networking

Το SDN αποτελείται από τρία βασικά επίπεδα: το Application Plane, το Control Plane και το Data Plane, επίπεδο εφαρμογών, επίπεδο ελέγχου και επίπεδο δεδομένων αντίστοιχα. Στην παρακάτω εικόνα φαίνεται αρχιτεκτονική των SDN δικτύων. [2]



Εικόνα 2.1: Αρχιτεκτονική SDN

Το Application Plane παρέχει στον χρήστη έναν τρόπο να προσπελάσει μέσω μιας εφαρμογής τις πληροφορίες ενός δικτύου. Αυτό το επίπεδο είναι η κύρια διασύνδεση του χρήστη με την εφαρμογή και, συνεπώς, με το δίκτυο. Στο επίπεδο αυτό γίνεται η διαχείριση των κατανεμημένων εφαρμογών. Στο επίπεδο αυτό υπάρχουν οι πολιτικές τους δικτύου μέσω των οποίων ο διαχειριστής του δικτύου μπορεί άμεσα να επικοινωνήσει με τον ελεγκτή ώστε να μάθει τις απαιτήσεις του δικτύου και να ορίσει την απαιτούμενη συμπεριφορά που πρέπει να εφαρμοστεί στο δίκτυο.

Στο Control Plane υπάρχει ο SDN ελεγκτής (SDN controller) ο οποίος επικοινωνεί με όλες τις δικτυακές συσκευές, λαμβάνει τις απαιτήσεις του συστήματος και ενημερώνει το επίπεδο εφαρμογών χρησιμοποιώντας την βόρια διεπαφή (Northbound interface).

Η βόρια διεπαφή (Northbound interface) δίνει την δυνατότητα στον διαχειριστή να έχει πρόσβαση στον ελεγκτή και να τον παραμετροποιήσει στις ανάγκες του συστήματος χρησιμοποιώντας εντολές, ακόμα και μικρά προγράμματα (scripts).

Το Data Plane είναι το επίπεδο που βρίσκεται το υλικό (hardware) καθώς και όλες οι φυσικές τους διασυνδέσεις. Μέσω μια διεπαφής (Southbound interface) ο διαχειριστής μπορεί άμεσα να ελέγξει και να επεξεργαστεί τα δεδομένα του δικτύου.

Η νότια διεπαφή (Southbound interface) χρησιμοποιείται για τον προγραμματισμό του επιπέδου των δεδομένων των συσκευών. Δεν είναι μια φυσική διεπαφή αλλά επαφή λογισμικού (API) και η δημοφιλέστερη είναι το OpenFlow.

2.3 Το πρωτόκολλο OpenFlow

Το OpenFlow δημιουργήθηκε από το Πανεπιστήμιο του Stanford στις Η.Π.Α. και είναι έργο ανοιχτού κώδικα. Είναι το πρώτο πρωτόκολλο που δημιουργήθηκε για την επικοινωνία του επιπέδου ελέγχου με το επίπεδο δεδομένων σε SDN αρχιτεκτονικές. Επιτρέπει στους ελεγκτές να καθορίσουν την διαδρομή των πακέτων κατά μήκος του δικτύου καθώς και τη διαχείριση μεταγωγέων διαφορετικών κατασκευαστών απομακρυσμένα με την χρήση ελεύθερου λογισμικού. Χρησιμοποιεί την έννοια των ροών για τον προσδιορισμό της κίνησης βάση κανόνων που μπορούν να προγραμματιστούν από τον SDN ελεγκτή. [3]

Είναι το πρώτο πρωτόκολλο επικοινωνίας που επιτρέπει στο επίπεδο ελέγχου του δικτύου να ορίσει κανόνες προώθησης, με τους οποίους μπορούμε να χειριστούμε τις συσκευές. Με την χρήση του έχουμε άμεση πρόσβαση και διαχείριση στο επίπεδο προώθησης των router και των switch. Το OpenFlow εφαρμόζεται και στα δύο μέρη της σύνδεσης και χρησιμοποιεί την έννοια των ροών για τον προσδιορισμό της κίνησης βάση προκαθορισμένων κανόνων.

Σε αντίθεση με τα παραδοσιακά IP δίκτυα που η δρομολόγηση στο διαδίκτυο γίνεται βάσει των διευθύνσεων IP και δύο ροών με αρχικά και τερματικά σημεία που η πληροφορία ακολουθεί πάντα, στην αρχιτεκτονική SDN δεν συμβαίνει το ίδιο. Πλέον δύο ροές μπορεί να έχουν ίδια σημεία πηγής και προορισμού αλλά να ακολουθούν άλλη διαδρομή και με διαφορετική προτεραιότητα διότι έχουν διαφορετικές απαιτήσεις.

Δίκτυα αρχιτεκτονικής SDN που υποστηρίζουν OpenFlow μπορούν να αναπτυχθούν σε ήδη υπάρχοντα δίκτυα. Αρκετοί είναι οι κατασκευαστές υλικού, οι οποίοι προσφέρουν συσκευές που λειτουργούν με την χρήση του OpenFlow παράλληλα με το δικό τους λογισμικό. Παρακάτω παρατίθενται μερικά από τα πιο γνωστά switches που υποστηρίζουν το OpenFlow.

Κατασκευαστής	Μοντέλο Switch
HP	8200ZL, 6600, 6200 3500
IBM	G8264
Pronto	3290, 3780
NEC	PF5240, PF820

Παρατήρηση: Η Cisco δεν υποστηρίζει ακόμα το OpenFlow σε καμία της συσκευή

2.4 Τα δομικά μέρη του OpenFlow

Τα βασικά δομικά μέρη του πρωτοκόλλου OpenFlow είναι:

- Ο OpenFlow controller
- Οι OpenFlow μεταγωγείς που αποτελούνται από
 - Πόρτες
 - Πίνακες ροών
 - Ασφαλές κανάλι OpenFlow

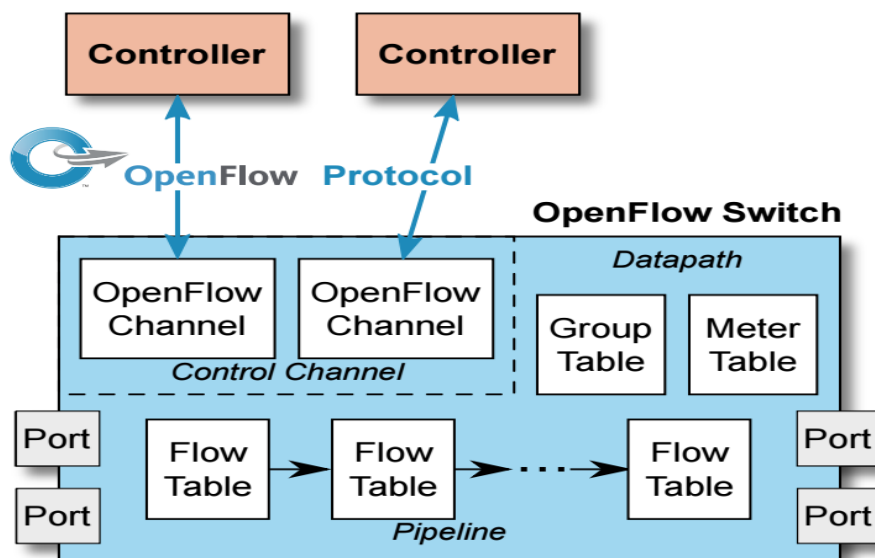


Figure 1: Main components of an OpenFlow switch.

Εικόνα 2.2: Δομή OpenFlow

2.4.1 OpenFlow Controller

Το OpenFlow είναι ένα μέσο επικοινωνίας μεταξύ ενός controller και του δικτυακού εξοπλισμού. Ένας ελεγκτής OpenFlow είναι ένας τύπος ελεγκτή SDN που χρησιμοποιεί το πρωτόκολλο OpenFlow. Χρησιμοποιεί το πρωτόκολλο OpenFlow για τη σύνδεση και τη διαμόρφωση των συσκευών δικτύου (δρομολογητές, μεταγωγείς κ.λπ.) και καθορίζει την καλύτερη διαδρομή για την κυκλοφορία της πληροφορίας.

Οι ελεγκτές SDN μπορούν να απλοποιήσουν τη διαχείριση δικτύου, να χειριστούν όλες τις επικοινωνίες μεταξύ εφαρμογών και συσκευών για να διαχειρίζονται και να τροποποιούν αποτελεσματικά τις ροές δικτύου για να ανταποκρίνονται στις μεταβαλλόμενες ανάγκες. Όταν το επίπεδο ελέγχου δικτύου πραγματοποιείται στο software και όχι στο firmware, οι διαχειριστές μπορούν να διαχειριστούν την κυκλοφορία του δικτύου πιο δυναμικά. Ένας ελεγκτής SDN μεταδίδει πληροφορίες στους δρομολογητές (μέσω των southbound APIs) και στις εφαρμογές (μέσω των northbound APIs).

Συγκεκριμένα, οι ελεγκτές OpenFlow δημιουργούν ένα κεντρικό σημείο ελέγχου για να εποπτεύουν μια ποικιλία εξαρτημάτων δικτύου με που υποστηρίζουν το πρωτόκολλο OpenFlow.[4]

2.4.2 OpenFlow μεταγωγέας (OpenFlow Switch)

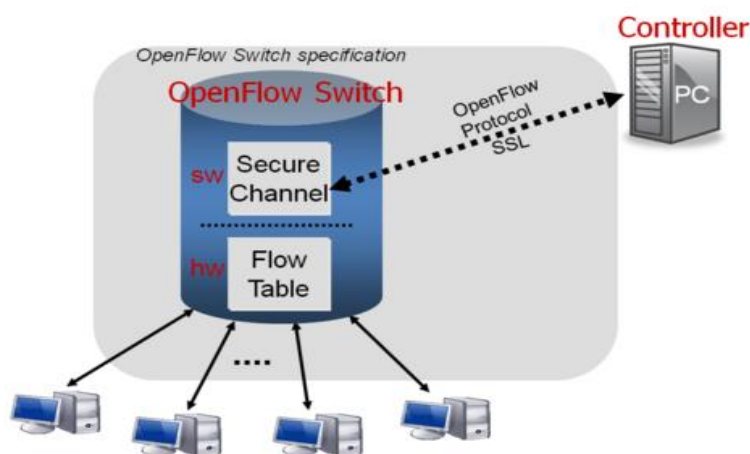
Ένας OpenFlow μεταγωγέας αποτελεί το εξοπλισμό που υποστηρίζει το πρωτόκολλο OpenFlow. Επικοινωνεί μέσω ενός OpenFlow κανάλι με έναν εξωτερικό ελεγκτή. Αναζητεί πακέτα και τα προωθεί σύμφωνα με έναν ή περισσότερους πίνακες ροής και έναν πίνακα ομάδας. Αποτελείται από δύο μέρη: το switch-agent και την πλατφόρμα δεδομένων.

Το switch-agent επικοινωνεί μέσω του OpenFlow πρωτοκόλλου με τον ελεγκτή καθώς και με την πλατφόρμα δεδομένων μέσω εσωτερικού κατάλληλου πρωτοκόλλου. Είναι υπεύθυνο στο να μεταφράζει τις εντολές του ελεγκτή σε γλώσσα μηχανής και να τις στέλνει στην πλατφόρμα δεδομένων προκειμένου να της είναι κατανοητές. Επίσης αντίστοιχα λαμβάνει από την πλατφόρμα δεδομένων

τυχόν ειδοποιήσεις και τις μετατρέπει σε OpenFlow μηνύματα κατανοητά από τον ελεγκτή και του τα προωθεί.

Η πλατφόρμα δεδομένων αναλαμβάνει τη διαχείριση και προώθηση των πακέτων, ενώ ανάλογα την παραμετροποίηση ενδέχεται να στείλει πακέτα στο switch-agent για επεξεργασία. Περιλαμβάνει πίνακες ροής και ανάλογες δράσεις για κάθε ροή, οπότε τα πακέτα προωθούνται ανάλογα σε ποια ροή ανήκουν και σε ποιον κανόνα του πίνακα ροής θα αντιστοιχιστούν.

Ένας OpenFlow μεταγωγέας μπορεί να λειτουργήσει μόνο με τη συνεργασία τριών βασικών στοιχείων: χρήση πινάκων ροής που είναι εγκατεστημένοι στους μεταγωγείς, έναν controller και το πρωτόκολλο OpenFlow, ώστε ο ελεγκτής να επικοινωνήσει με ασφάλεια με τους μεταγωγείς. Οι πίνακες ροής υπάρχουν στους μεταγωγείς. Οι ελεγκτές επικοινωνούν με τους μεταγωγείς μέσω του πρωτοκόλλου OpenFlow και επιβάλλουν πολιτικές στις ροές.[5]



Εικόνα 2.3: OpenFlow Switch

2.4.3 Πόρτες (Ports)

Οι πόρτες σε έναν OpenFlow μεταγωγέα λειτουργούν με παρόμοιο τρόπο όπως σε κάθε μεταγωγέα. Τα πακέτα εισέρχονται σε μία πόρτα εισόδου, τα επεξεργάζεται ο μεταγωγέας και στη συνέχεια μεταφέρονται σε μία πόρτα εξόδου. Στο OpenFlow υπάρχουν τριών ειδών πόρτες οι Φυσικές, οι Λογικές και οι Κλειστές.[3]

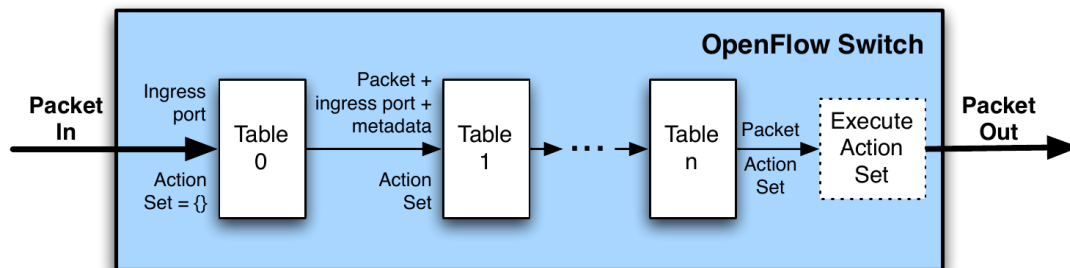
- **Φυσικές πόρτες:** Σας φυσικές πόρτες ορίζονται οι πόρτες του δικτυακού εξοπλισμού. Όταν στο υλικό υποστηρίζονται περισσότεροι του ενός εικονικοί μεταγωγείς τότε μια OpenFlow φυσική πόρτα αποτελεί μέρος του φυσικού εξοπλισμού.
- **Λογικές πόρτες:** Οι λογικές πόρτες δεν αντιστοιχούν στις φυσικές πόρτες του εξοπλισμού. Το OpenFlow είναι ανεξάρτητο από το πώς αντιστοιχίζονται σε φυσικές πόρτες γι' αυτό και αυτό τις αντιμετωπίζει παρόμοια με τις φυσικές.
- **Κλειστές πόρτες:** Είναι οι πόρτες που χρησιμοποιούνται για την εσωτερική διαχείριση των πακέτων. Χωρίζονται σε 5 κατηγορίες:
 - **ALL:** Είναι όλες οι πόρτες που μπορούν να χρησιμοποιηθούν σαν εξωτερικές.
 - **Controller:** Είναι η πόρτα όπου το switch επικοινωνεί μέσω του ασφαλούς καναλιού με τον controller.
 - **Table:** Είναι μια εισερχόμενη πόρτα
 - **IN_PORT:** Είναι εισερχόμενη και εξερχόμενη πόρτα ταυτόχρονα
 - **ANY:** Μπορεί να χρησιμοποιηθεί σαν εισερχόμενη, εξερχόμενη η οποιαδήποτε άλλη ανάλογα με τις απαιτήσεις του συστήματος.

2.4.4 OpenFlow Πίνακες (OpenFlow tables)

Με την μεταφορά μηνμάτων μεταξύ του controller και του μεταγωγέα δημιουργούνται πίνακες έτσι ώστε οι μεταγωγείς να μπορούν να διαχειρίζονται τα πακέτα. Οι πίνακες αποτελούνται από τα εξής πεδία:

- **Match:** Σε αυτό το πεδίο αναγράφεται η πόρτα εισόδου και η επικεφαλίδα του πακέτου.
- **Priority:** Εδώ καθορίζεται η προτεραιότητα της εγγραφής.
- **Counter:** Αναγράφονται τα στατιστικά στοιχεία των πακέτων.
- **Instruction:** Καθορίζεται τις ενέργειες του μεταγωγέα.
- **Timeouts:** Αναφέρεται ο μέγιστος χρόνος ισχύς της εγγραφής.

- **Flags:** Χρησιμοποιείται για την αλλαγή του τρόπου διαχείρισης των εγγραφών.



Εικόνα 2.4: OpenFlow πίνακες

2.4.5 Ασφαλές Κανάλι (OpenFlow Channel)

Για την επικοινωνία μεταξύ controller και εξοπλισμού απαιτείται ένα κανάλι όπου θα μεταφέρονται όλα τα μηνύματα και τα πακέτα που αφορούν την λειτουργία του δικτύου. Οι επικοινωνία μεταξύ τους κρυπτογραφείται μέσω TLS πρωτοκόλλου.

Όταν πραγματοποιείται η επικοινωνία, ο controller και ο μεταγωγέας στέλνουν μεταξύ τους μηνύματα για να «αποφασίσουν» ποια έκδοση του OpenFlow θα χρησιμοποιήσουν. Θα χρησιμοποιηθεί η νεότερη έκδοση που υποστηρίζουν και οι δυο πλευρές. Αν δεν υποστηρίζεται καμία κοινή έκδοση τότε στέλνονται μηνύματα σφάλματος και η σύνδεση τερματίζεται και ο μεταγωγέας εισέρχεται σε μία από τις παρακάτω καταστάσεις:

- **Fail Secure Mode:** Διακόπτεται η επικοινωνία με τον controller και χρησιμοποιούνται οι υπάρχοντες εγγραφές των πινάκων για την διαχείριση των πακέτων μέχρι να λήξει η ισχύ τους.
- **Fail Standalone Mode:** όπου ο μεταγωγέας επανέρχεται στην κανονική του λειτουργία.

Η αξιοπιστία της αποστολής και λήψης των μηνυμάτων βασίζεται αποκλειστικά στον μηχανισμό TLS και στην TCP σύνδεση καθώς δεν υφίσταται κάποια λειτουργία επιβεβαίωσης λήψης ούτε και διατήρησης της σωστής σειράς.

3. Open Daylight Controller

Ο SDN ελεγκτής αποτελεί τη μονάδα ελέγχου του ευφυούς δικτύου, αντιπροσωπεύοντας το επίπεδο ελέγχου μετά το διαχωρισμό του από το επίπεδο δεδομένων. Είναι η πλατφόρμα που εποπτεύει και διαχειρίζεται τις ροές κίνησης μέσω του ελέγχου των δικτυακών συσκευών με τα southbound APIs και παράλληλα ενημερώνει και διαχειρίζεται τις εφαρμογές με τα northbound APIs.

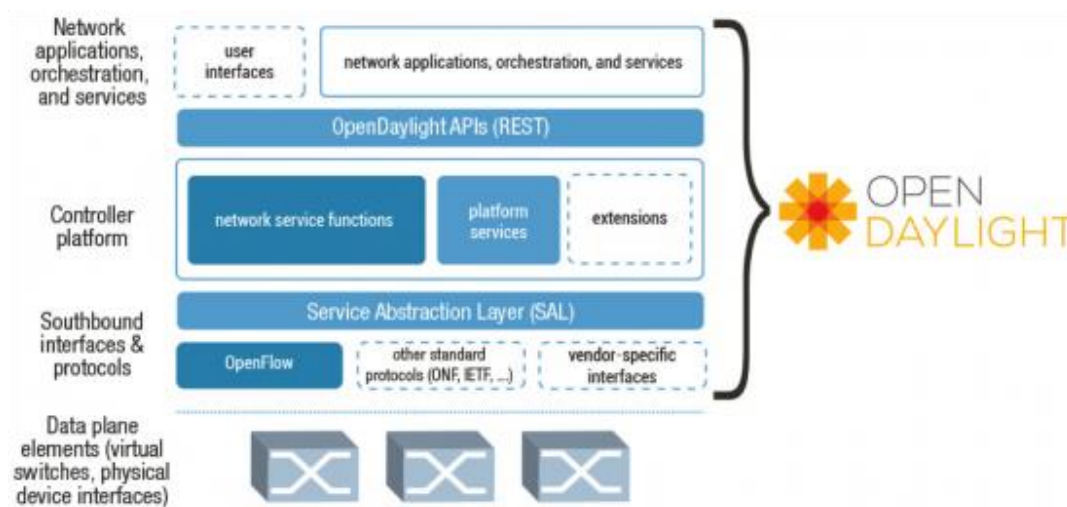
Οι SDN ελεγκτές χρησιμοποιούν συνήθως το OpenFlow πρωτόκολλο για την επικοινωνία τους με τις δικτυακές συσκευές. Επίσης υπάρχουν και άλλα πρωτόκολλα γι' αυτήν την επικοινωνία όμως στην παρούσα διπλωματική επιλέχθηκε ο OpenDaylight SDN ελεγκτής.

Ο OpenDaylight ελεγκτής αναπτύχθηκε από την κοινότητα Linux Foundation και αποτελεί το μεγάλο έργο Ανοικτού Λογισμικού σχετικά με το Software Defined Networking και ο κώδικάς του είναι γραμμένος σε γλώσσα JAVA. Ο OpenDaylight υποστηρίζει το OpenFlow πρωτόκολλο καθώς και άλλα πρωτόκολλα και τεχνολογίες SDN αρχιτεκτονικής.[6]

3.1 Αρχιτεκτονική του OpenDaylight

Ο Open Daylight ελεγκτής αποτελείται από τρία μέρη:

- Την πλατφόρμα του ελεγκτή
- Τα Northbound APIs
- Τα Southbound APIs



Εικόνα 3.1: Open Daylight

3.2 Πλατφόρμα ελεγκτή

Η πλατφόρμα του ελεγκτή εμπεριέχει τόσο Northbound όσο και Southbound διεπαφή.

Η Northbound διεπαφή παρέχει στον ελεγκτή υπηρεσίες καθώς και ένα σύνολο REST API εφαρμογών που μπορούν να χρησιμοποιηθούν για τη διαχείριση και την παραμετροποίηση του δικτύου.

Η Southbound διεπαφή υλοποιεί πρωτόκολλα για τη διαχείριση και τον έλεγχο της δικτυακής υποδομής που βρίσκεται στο κατώτερο επίπεδο. Εδώ υλοποιούνται αρκετές επεκτάσεις είτε για την υποστήριξη δικτυακών πρωτοκόλλων είτε για την απευθείας επικοινωνία με το υλικό. Τα πιο γνωστά πρωτόκολλα που υλοποιούνται είναι τα OpenFlow, NetConf, SNMP.

Η πλατφόρμα του ελεγκτή επικοινωνεί με το δίκτυο χρησιμοποιώντας τη Southbound διεπαφή και παρέχει βασικές δικτυακές υπηρεσίες μέσω ενός συνόλου λειτουργιών: Host Tracker, L2 Switch, OpenFlow Forwarding Rules Manager, OpenFlow Stats Manager, OpenFlow Switch Manager, Topology Processing.

3.2.1 Base Network Service Functions

➤ **Host Tracker**

Αποθηκεύει πληροφορίες για τα τερματικά του συστήματος (διευθύνσεις MAC, IP, port number) και παρέχει API για την ανάκτηση πληροφοριών.

Δύναται να δουλέψει με στατικό ή δυναμικό τρόπο. Στατικώς, η βάση δεδομένων του Host Tracker μεταχειρίζεται χειροκίνητα μέσω Northbound APIs ενώ δυναμικώς ο Host Tracker χρησιμοποιεί το ARP πρωτόκολλο για να ανιχνεύσει τις πληροφορίες για τη βάση δεδομένων του.

➤ **L2 Switch**

Με την άφιξη ενός πακέτου, η λειτουργία αυτή μαθαίνει την MAC διεύθυνση της πηγής.

Εάν γνωρίζει τον προορισμό, μεταφέρει το πακέτο στον προορισμό, διαφορετικά στέλνει broadcast μήνυμα (δηλαδή μήνυμα προς όλες τις συσκευές) σε όλες τις εξωτερικές πόρτες του δικτύου.

➤ **OpenFlow Forwarding Rules Manager**

Διαχειρίζεται βασικούς κανόνες προώθησης , επιλύει τυχόν συγκρούσεις μεταξύ των κανόνων αυτών και τους επικυρώνει. Επικοινωνεί με Southbound επεκτάσεις φορτώνει OpenFlow κανόνες στους διαχειριζόμενους μεταγωγείς.

➤ **OpenFlow Statistics Manager**

Υλοποιεί τη συλλογή στατιστικών αποστέλλοντας αιτήσεις για στατιστικά σε όλους τους ενεργούς κόμβους (δηλαδή τους διαχειριζόμενους μεταγωγείς) του ευφυούς δικτύου και αποθηκεύει τις απαντήσεις τους στο operational πεδίο. Επίσης επιστρέφει πληροφορίες στο Northbound API για τα εξής:

- flow
- meter
- table
- group statistics

➤ **OpenFlow Switch Manager**

Παρέχει πληροφορίες για τους κόμβους (μεταγωγείς) τους ευφυούς δικτύου καθώς και τις πόρτες στις οποίες συνδέονται. Όσο ο ελεγκτής ανακαλύπτει νέα δικτυακά στοιχεία, οι 26 πληροφορίες γι' αυτά αποθηκεύονται στο δέντρο δεδομένων του Switch Manager. Μπορεί να χρησιμοποιηθεί το Northbound API για την ανάκτηση των πληροφοριών αυτών.

➤ **Topology Processing**

Αποθηκεύει και μεταχειρίζεται πληροφορίες σχετικά με τις διαχειριζόμενες δικτυακές συσκευές. Όταν ο ελεγκτής εκκινείτε, ο Topology Manager δημιουργεί τον κεντρικό κόμβο της τοπολογίας. Έκτοτε ενημερώνεται για την τοπολογία, για τυχόν ειδοποιήσεις και αλλαγές σ' αυτό το δέντρο, συμπεριλαμβανομένων όλων των μεταγωγέων που έχουν ανακαλυφθεί.

3.3 Rest Αρχιτεκτονική

Το REST (Representational State Transfer) είναι μια αρχιτεκτονική που χρησιμοποιείται σε διαδικτυακές εφαρμογές. Τα συστήματα που σχεδιάζονται σε REST αρχιτεκτονική έχουν καλή επίδοση, αξιοπιστία και δυνατότητα επέκτασης ως

προς το ποσοστό των χρηστών που μπορούν να υποστηρίξουν. Τα RESTful συστήματα επικοινωνούν μέσω του HTTP πρωτοκόλλου και των μεθόδων GET, PUT, POST και DELETE τις οποίες χρησιμοποιεί στείλει δεδομένα σε απομακρυσμένους εξυπηρετητές.

Τα πλεονεκτήματα της REST αρχιτεκτονικής σε σχέση με τη SOAP είναι :

- Μπορεί να αξιοποιηθεί η cache μνήμη
- Πρόκειται για stateless πρωτόκολλο, δηλαδή υπάρχει ανεξαρτησία μεταξύ της κάθε αίτησης πελάτη και αντίστοιχης απάντησης
- Αξιοποιείται ένα πολυεπίπεδο σύστημα
- Αξιοποιείται μια ενιαία διεπαφή

Το πρωτόκολλο της REST αρχιτεκτονικής είναι το RESTCONF. Το RESTCONF επιτρέπει πρόσβαση σε δομές δεδομένων στον ελεγκτή. Υπάρχουν δύο ειδών δομές δεδομένων:

- Config: Περιέχει τις πληροφορίες που προστίθενται μέσω του ελεγκτή
- Operational: Περιέχει τις πληροφορίες που προστίθενται μέσω του δικτύου

Ο OpenDaylight έχει REST αρχιτεκτονική οπότε μπορεί να προσφέρει όλα τα παραπάνω.[7]

Κάθε αίτηση που του γίνεται μέσω HTTP μεθόδου, πρέπει να γίνεται στο αντίστοιχο URI, που έχει τη μορφή :

<http://<odl-IP>:8181/restconf/operational/<path>/>

<http://<odl-IP>:8181/restconf/config/<path>/>

3.4 Εγκατάσταση OpenDaylight

Για την εγκατάσταση του OpenDaylight είναι απαραίτητη η λήψη της νεότερης έκδοσης καθώς και η ύπαρξη των κατάλληλων βιβλιοθηκών JAVA στο μηχάνημα. Η εγκατάστασή τους γίνεται με τις παρακάτω εντολές:

- `sudo apt-get install maven openjdk-7-jdk openjdk-7-jre`
- `export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64`

Στη συνέχεια αφού εγκατασταθούν οι απαραίτητες βιβλιοθήκες θα χρειαστεί να κατεβάσουμε την εφαρμογή από την διεύθυνση <http://www.opendaylight.org/software/downloads> και να κάνουμε τρία ακόμα βήματα:

- Κάνουμε unzip το αρχείο που κατεβάσαμε
- Πηγαίνουμε στο directory στο οποίο έχει κατέβει το αρχείο χρησιμοποιώντας την εντολή cd
- Τέλος για να τρέξουμε την εφαρμογή χρησιμοποιούμε την εντολή `./bin/karaf`

```
brian@odl:~$ cd distribution-karaf-0.4.0-Beryllium/
brian@odl:~/distribution-karaf-0.4.0-Beryllium$ ./bin/karaf
```



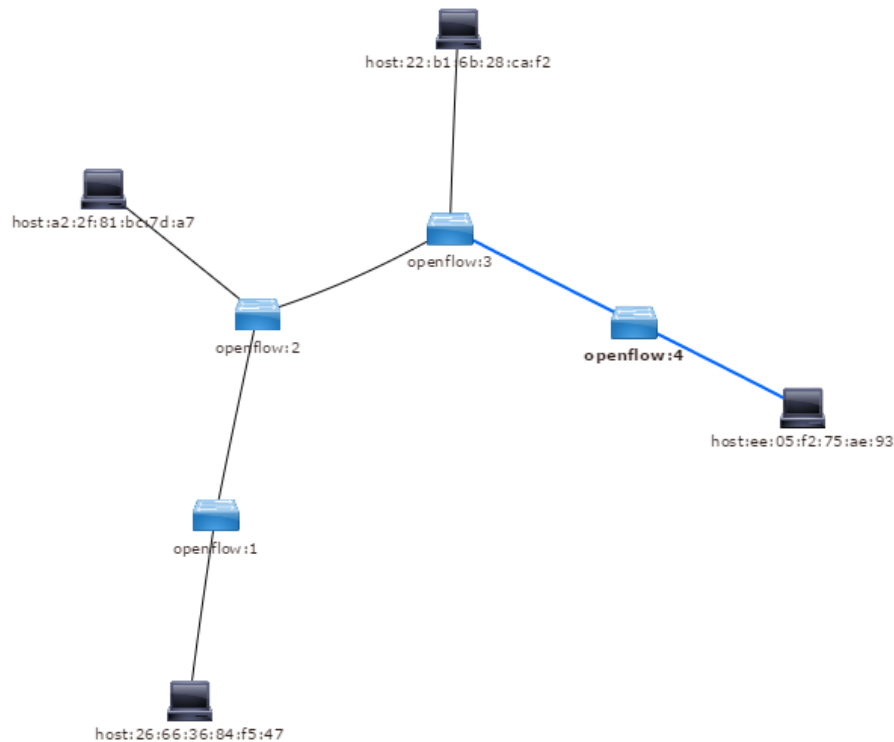
```
Hit '<tab>' for a list of available commands
and '[cnd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown OpenDaylight.

opendaylight-user@root>
```

Εικόνα 3.2: Open Daylight εγκατάσταση

3.5 Mininet

Το Mininet είναι ένας εξομοιωτής δικτύου. Τρέχει μια συλλογή από end-hosts, switches, routers, και ζεύξεις σε έναν απλό Linux πυρήνα. Χρησιμοποιεί virtualization για να κάνει ένα απλό σύστημα να μοιάζει με ολόκληρο δίκτυο που τρέχει στον ίδιο πυρήνα και σύστημα.[8] Ένας Mininet host συμπεριφέρεται όπως ακριβώς μια πραγματική μηχανή, μπορείς να κάνεις ssh σε αυτόν και να τρέξεις αυθαίρετα προγράμματα (συμπεριλαμβανομένου οτιδήποτε είναι εγκατεστημένο κάτω από το σύστημα των Linux). Τα προγράμματα που τρέχεις μπορούν να στέλνουν πακέτα μέσω διεπαφή που μοιάζει με την πραγματική Ethernet διεπαφή, με ταχύτητα ζεύξης και καθυστέρηση. Τα πακέτα επεξεργάζονται από αυτό που μοιάζει με αληθινό Ethernet switch ή router με συγκεκριμένο ποσό πακέτων στην ουρά. Όταν δύο προγράμματα επικοινωνούν μέσω του Mininet, η μετρήσιμη απόδοση πρέπει να ταιριάζει αυτή του native μηχανήματος.[9]



Εικόνα 3.3: Mininet Topology

Σε συντομία, οι εικονικοί host του Mininet, switches, links, και controllers είναι οι πραγματικοί –απλά δημιουργούνται με την χρήση λογισμικού και όχι υλικού – και το μεγαλύτερο μέρος της συμπεριφοράς τους είναι παρόμοιο με αυτό των συσκευών υλικού.

Υπάρχουν πολλοί λόγοι για να επιλέξουμε το mininet σαν εργαλείο εξομοίωσης. Πρώτον, υπάρχουν λίγες συσκευές δικτύου με σκοπό την εφαρμογή του SDN standard καθώς δεν είναι μια ευρέως διαδεδομένη η συγκεκριμένη τεχνολογία. Επιπλέον η εφαρμογή ενός δικτύου με μεγάλο αριθμό συσκευών είναι κοστοβόρα υπόθεση.

Μερικά από τα Θετικά του Mininet

- Είναι γρήγορο να ξεκινήσεις ένα απλό δίκτυο σε λίγα μόνο δευτερόλεπτα.
- Μπορούμε να δημιουργήσουμε όποιες τοπολογίες επιθυμούμε: ένα απλό switch, μεγαλύτερες τοπολογίες, κέντρο δεδομένων ή οτιδήποτε άλλο.

- Μπορούμε να τρέξουμε πραγματικά προγράμματα: οτιδήποτε τρέχει σε Linux είναι διαθέσιμο, όπως web servers σε TCP window monitoring ακόμα και εργαλεία όπως το Wire shark.
- Μπορούμε να τροποποιήσουμε τον τρόπο προώθησης των πακέτων: Τα switches του Mininet είναι προγραμματιζόμενα με την χρήση του OpenFlow πρωτοκόλλου.
- Μπορούμε να μοιραστούμε και να αναπαράγουμε αποτελέσματα.
- Το Mininet είναι open source.

3.6 Χαρακτηριστικά Mininet

Παρακάτω παρουσιάζονται μερικά χαρακτηριστικά του Mininet:

- Ευελιξία, νέες τοπολογίες και νέα χαρακτηριστικά μπορούν να τοποθετούνται στο λογισμικό, χρησιμοποιώντας γλώσσες προγραμματισμού και κοινά λειτουργικά συστήματα.
- Διαδραστικότητα. Η διαχείριση και η εκτέλεση του εξομοιωμένου δικτύου πρέπει να συμβαίνει σε πραγματικό χρόνο, όπως συμβαίνει στα πραγματικά δίκτυα.
- Ρεαλισμός, καθώς η συμπεριφορά του πρωτότυπου πρέπει να αντιπροσωπεύει την πραγματική συμπεριφορά με μεγάλο βαθμό εμπιστοσύνης, έτσι ώστε οι εφαρμογές και τα πρωτόκολλα να χρησιμοποιούνται χωρίς μεταβολές του κώδικα.
- Τα πρωτότυπα δίκτυα που δημιουργούνται πρέπει εύκολα να μοιράζονται με άλλους, οι οποίοι μπορούν να τρέξουν και να τροποποιήσουν τα πειράματα.
- Scalability, τα πρότυπα δίκτυα που δημιουργούνται πρέπει να μπορούν να εφαρμόζονται σε μεγάλη κλίμακα με εκατοντάδες switches και στοιχεία δικτύου.

3.7 Περιορισμοί του Mininet

Το Mininet είναι πολύ χρήσιμο αλλά έχει και περιορισμούς στην χρήση του. Παρακάτω παρουσιάζονται οι πιο σημαντικοί περιορισμοί για την ορθή λειτουργία του:

- Περιορισμός πόρων τρέχοντος συστήματος. Για παράδειγμα εάν ο server έχει 3 GHz CPU και μπορεί να κάνει μεταγωγή 10 Gbps προσομοιωμένης κίνησης, αυτοί οι πόροι θα πρέπει να μοιραστούν και να ισορροπήσουν μεταξύ των virtual hosts και switches.
- Το Mininet χρησιμοποιεί έναν απλό Linux πυρήνα για όλους τους εικονικούς hosts. Αυτό σημαίνει ότι μπορούμε να τρέχουμε λογισμικό βασισμένο σε BSD, Windows ή άλλο λειτουργικό σύστημα.
- Το Mininet δίκτυο είναι απομονωμένο από το LAN και από το internet. Παρόλα αυτά μπορούμε να χρησιμοποιήσουμε τη NAT επιλογή για να συνδέσουμε το δίκτυο Mininet στο LAN μέσω Network Address Translation. Μπορούμε επίσης να προσδέσουμε πραγματική ή εικονική διεπαφή υλικού στο Mininet δίκτυο.

3.8 Εγκατάσταση Mininet

Για την εγκατάσταση του Mininet θα χρειαστεί να τρέξουμε μια σειρά από εντολές οι οποίες περιγράφονται παρακάτω:

- Αρχικά διαγράφουμε τυχόν υπολείμματα από παλαιότερες εκδόσεις του Mininet ή του Open Vswitch:

```
sudo rm -rf /usr/local/bin/mn /usr/local/bin/mnexec \
/usr/local/lib/python*/*/mininet* \ /usr/local/bin/ovs-*
/usr/local/sbin/ovs-*
```

- Έπειτα ελέγχουμε την έκδοση Linux που έχουμε:

```
lsb_release -a
```

- Μετά εγκαθιστούμε το Mininet για την έκδοση Linux που τρέχουμε:

```
Mininet 2.1.0 on Ubuntu 14.10: sudo apt-get install mininet
```

```
Mininet 2.1.0 on Ubuntu 14.04: sudo apt-get install mininet
```

```
Mininet 2.0.0 on Ubuntu 12.04: sudo apt-get install mininet/precise-backports
```

Χρησιμοποιούμε μια από τις παραπάνω εντολές.

- Στην συνέχεια πρέπει να απενεργοποιήσουμε το openvswitch-controller:

```
sudo service openvswitch-controller stop
```

```
sudo update-rc.d openvswitch-controller disable
```

- Τέλος τεστάρουμε ότι το Mininet τρέχει σωστά:

```
sudo mn --test pingall
```

Αν μετά την τελευταία εντολή το Mininet «παραπονεθεί» ότι το Open vSwitch δεν λειτουργεί τότε τρέχουμε τις παρακάτω δύο εντολές:

```
sudo dpkg-reconfigure openvswitch-datapath-dkms
```

```
sudo service openflow-switch restart
```

Τέλος για να αποκτήσουμε και διάφορες έξτρα δυνατότητες που προσφέρει το Mininet όπως το Openflow reference switch, το reference controller και το Wireshark η χρήση την συγκεκριμένης εντολής είναι απαραίτητη:

```
git clone git://github.com/mininet/mininet
```

```
mininet/util/install.sh -fw
```

3.9 Αναβάθμιση παλαιότερης έκδοσης Mininet

Για την αναβάθμιση του προγράμματος Mininet από παλαιότερη σε μια πιο νέα έκδοση εκτελούμε τις εξής εντολές:

```
Cd mininet
```

```
Git fetch
```

Git checkout master
Git pull
Sudo make install

3.10 Έναρξη τοπολογίας

Για την εκτέλεση script στο Mininet, εκτελείται η εντολή:

```
sudo mn --custom topo.py --topo topology --controller=remote, ip=<odl-IP>
```

όπου καθορίζεται η τοπολογία που θα τρέξει, η ονομασία του script στο οποίο βρίσκεται καθώς και η σύνδεση της τοπολογίας με τον ελεγκτή μέσω της διεύθυνσής του.

```
mininet@mininet-vm:~$ sudo mn --topo single,8 --mac --switch ovsk --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1) (h7, s1) (h8, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8
*** Starting controller
*** Starting 1 switches
s1
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X X X X X
h2 -> X X X X X X X
h3 -> X X X X X X X
h4 -> X X X X X X X
h5 -> X X X X X X X
h6 -> X X X X X X X
h7 -> X X X X X X X
h8 -> X X X X X X X
*** Results: 100% dropped (0/56 received)
mininet>
```

Εικόνα 3.4: Εκτέλεση script

4. Data Centers

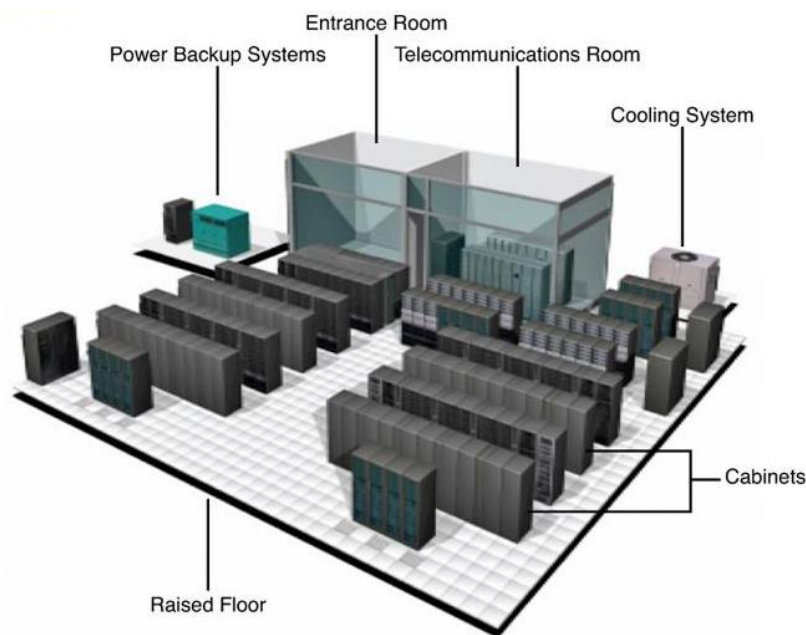
Τα πρώτα χρόνια της ανάπτυξης της τεχνολογίας της πληροφορικής, τα κέντρα δεδομένων αποτελούσαν τεράστιες αίθουσες με ηλεκτρονικούς υπολογιστές. Τα πρώτα υπολογιστικά συστήματα ήταν πολύ δύσκολο να συντηρηθούν και να διαχειριστούν καθώς απαιτούσαν ένα ειδικό περιβάλλον λειτουργίας. Ήταν απαραίτητα πολλά καλώδια για να συνδεθούν όλα τους εξαρτήματα όπως επίσης racks και υπερυψωμένα πατώματα. Η ασφάλεια ήταν επίσης ένα σημαντικό θέμα διότι τα πρώτα κέντρα δεδομένων χρησιμοποιούνταν για στρατιωτικούς και κυβερνητικούς σκοπούς.[10]

4.1 Εξέλιξη των Κέντρων δεδομένων στο πέρασμα των ετών

Κατά την διάρκεια της άνθησης της μικροϋπολογιστικής βιομηχανίας, η οποία παρατηρείται την διάρκεια του 1980, οι υπολογιστές άρχισαν να αναπτύσσονται παντού. Παράλληλα αναπτυσσόταν και η τεχνολογία της πληροφορικής (IT) μέχρι που την δεκαετία του '90 έγινε η έλευση του μοντέλου πελάτη-εξυπηρετητή (client-server) και των μικροϋπολογιστών (servers). Παράλληλα με τις παραπάνω τεχνολογίες η εμπορική ανάπτυξη του διαδικτύου (internet) έφερε την έκρηξη στην χρήση των κέντρων δεδομένων. Από τότε οι εταιρίες χρειαζόντουσαν γρήγορη σύνδεση στο διαδίκτυο (internet) και ασταμάτητη λειτουργία όχι μόνο για την ανάπτυξη των συστημάτων τους αλλά και για την εγκαθίδρυση της παρουσίας τους στο διαδίκτυο. Γι' αυτό τον λόγο η χρήση των κέντρων δεδομένων γνώρισε μεγάλη ανάπτυξη.

Η σύγχρονη μορφή των κέντρων δεδομένων έχει τη μορφή μεγάλων εγκαταστάσεων που χρησιμοποιούνται για να στεγάσουν υπολογιστικά συστήματα, όπως υπολογιστές, εξυπηρετητές (servers), διακομιστές ιστοσελίδων (webservers), διακομιστές εφαρμογών (application servers), διακομιστές αλληλογραφίας (mail servers), μεταγωγείς (switches), δρομολογητές (routers), συσκευές αποθήκευσης δεδομένων και γενικότερα τέτοιου είδους εξοπλισμό. Στην ουσία, το κέντρο

δεδομένων αποτελεί μια κεντρική αποθήκη, φυσική ή εικονική, για αποθήκευση, διαχείριση, διάδοση δεδομένων και ανταλλαγή ψηφιακών πληροφοριών που οργανώνεται γύρω από μια συγκεκριμένη επιχείρηση. Οι υπηρεσίες που παρέχουν συνήθως είναι η φιλοξενία ιστοσελίδων στο διαδίκτυο, intranet, υπηρεσίες που αφορούν τις τηλεπικοινωνίες και την τεχνολογία της πληροφορίας. Αναπόσπαστα συστατικά ενός κέντρου δεδομένων αποτελούν συνήθως οι συνδέσεις δικτύων, εφεδρικές και μη παροχές ρεύματος, έλεγχοι περιβάλλοντος (π.χ. κλιματισμός, πυρασφάλεια) καθώς και συστήματα ασφαλείας.



Εικόνα 4.1: Συστατικά κέντρων δεδομένων

Η βασική δομή των κέντρων δεδομένων αποτελείται από τρία επίπεδα, όπως παρουσιάζεται παρακάτω. Το επίπεδο πυρήνα αποτελεί τη ραχοκοκαλιά του δικτύου και περιέχει του τελικούς μεταγωγείς και μεγάλης ταχύτητας καλώδια όπως είναι οι οπτικές ίνες. Το επίπεδο αυτό δεν καθορίζει ούτε διευθύνει την κυκλοφορία του LAN. Επιπλέον, οι συσκευές αυτού του επιπέδου δεν πραγματοποιούν τον χειρισμό των πακέτων. Αντίθετα, το επίπεδο αυτό συνδέεται με την ταχύτητα και εξασφαλίζει την αξιόπιστη μεταφορά των πακέτων. Μερικά συστατικά του επιπέδου αυτού είναι οι μεταγωγείς για χρήση WAN-LAN, frame-relay συνδέσεις, ATM δίκτυα και Switched Multimanaged Data Services (SMDS).

Το επίπεδο συνάθροισης περιλαμβάνει δρομολογητές και μεταγωγείς που βασίζονται σε LAN τριών επιπέδων. Το επίπεδο αυτό εξασφαλίζει ότι τα πακέτα διευθετούνται σωστά ανάμεσα στα υποδίκτυα και στα VLANs της επιχείρησης. Οι υπηρεσίες που προσφέρει αυτό το επίπεδο είναι:

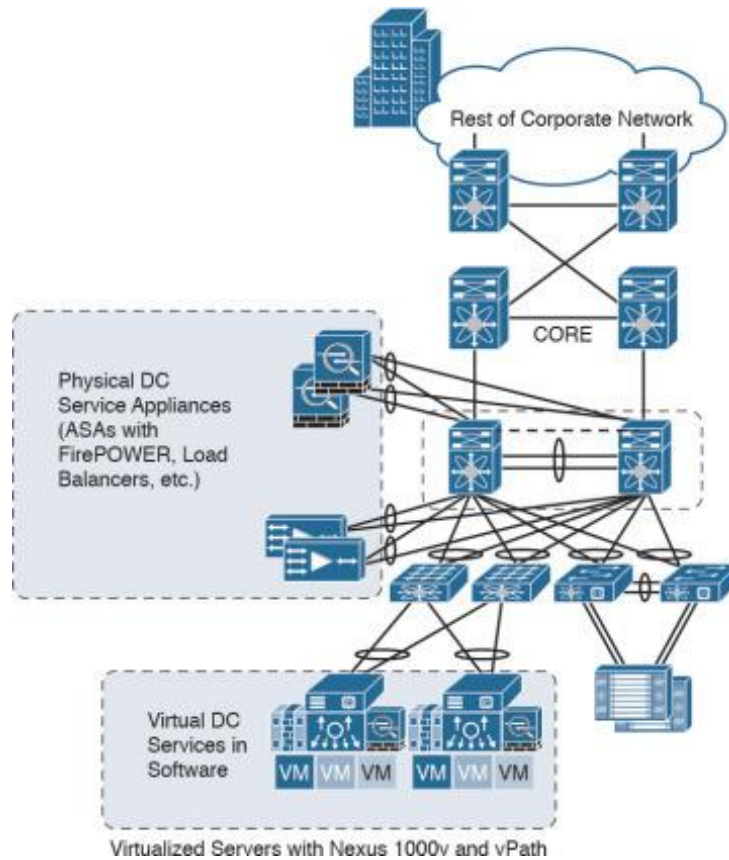
- Φιλτράρισμα πακέτων (firewalling): Επεξεργάζεται τα πακέτα και ρυθμίζει την μετάδοση τους με βάση την πηγή και τον προορισμό της πληροφορίας.
- QoS: Ο δρομολογητής ή οι μεταγωγείς μπορούν να διαβάσουν τα πακέτα και να προτεραιοποιήσουν την παράδοση τους, με βάση τις πολιτικές έχουν οριστεί.
- Παρέχει τον έλεγχο της μετάδοσης σε όλους τους υπολογιστές του υποδικτύου (broadcast) και της πολλαπλής διανομής (multicast).
- Επιτρέπει τη χρήση πρωτοκόλλων στις θύρες από και προς διαφορετικές αρχιτεκτονικές δικτύων.
- Το επίπεδο αυτό παρέχει επίσης διευθέτηση των πακέτων στην κίνηση του δικτύου.

Το επίπεδο πρόσβασης αποτελείται από διανομείς και μεταγωγείς και εστιάζει στη σύνδεση των κόμβων. Το επίπεδο αυτό εξασφαλίζει ότι τα πακέτα διανέμονται στους τελικούς χρήστες-υπολογιστές. Αποτελείται από υπολογιστές (desktops και laptops) και εξυπηρετητές όπως web servers, ftp servers, database servers κλπ.

Τα προτερήματα του συγκεκριμένου μοντέλου είναι:

- Υψηλή απόδοση: Προσφέρει δίκτυα υψηλής απόδοσης, όπου μόνο ορισμένα στρώματα εμφανίζουν συμφόρηση.
- Αποτελεσματική διαχείριση και αντιμετώπιση προβλημάτων: Δίνει τη δυνατότητα αποτελεσματικής διαχείρισης του δικτύου απομόνωσης των αιτών των προβλημάτων του δικτύου.
- Δημιουργία πολιτικής: Εύκολη δημιουργία κανόνων και πολιτικών.
- Δυνατότητα επέκτασης: Εύκολη ανάπτυξη με τη διαίρεση του δικτύου σε λειτουργικές περιοχές.

- Πρόβλεψη συμπεριφοράς: Κατά το σχεδιασμό ή τη διαχείριση ενός δικτύου, το μοντέλο αυτό επιτρέπει να προσδιοριστεί τι θα συμβεί με το δίκτυο, εάν νέες πιέσεις προστεθούν σε αυτό.



Εικόνα 4.2: Δομή κέντρου δεδομένων

4.2 Απαιτήσεις Κέντρων Δεδομένων

Οι λειτουργίες των υπολογιστών είναι η πιο κρίσιμη πτυχή των οργανωτικών λειτουργιών. Ένα από τα πιο βασικά προβλήματα είναι η συνέχεια της επιχειρησιακής δραστηριότητας καθώς οι εταιρίες βασίζονται στα πληροφοριακά τους συστήματα για την διεύθυνση των εργασιών τους. Εάν κάποιο σύστημα δεν είναι πλέον διαθέσιμο, οι λειτουργίες της εταιρίας μπορεί να μειθούν ή και να σταματήσουν εντελώς. Είναι αναγκαίο να παρέχεται μια αξιόπιστη υποδομή για τις IT λειτουργίες, έτσι ώστε να ελαχιστοποιηθεί η οποιαδήποτε πιθανότητα διακοπής. Η ασφάλεια των συστημάτων και η ασφάλεια των πληροφοριών είναι επίσης σημαντικές, για τον λόγο αυτό ένα κέντρο δεδομένων πρέπει να προσφέρει ένα ασφαλές περιβάλλον που να ελαχιστοποιεί τις πιθανότητες παραβίασης της

ασφάλειας. Επομένως ένα κέντρο δεδομένων πρέπει να διατηρεί υψηλά πρότυπα για το υπολογιστικό περιβάλλον που φιλοξενεί.

Η αποτελεσματική λειτουργία ενός κέντρου δεδομένων απαιτεί την ισορροπημένη επένδυση τόσο στις εγκαταστάσεις όσο και την στέγαση του εξοπλισμού.

Υπάρχει μια τάση για τον εκσυγχρονισμό των κέντρων δεδομένων, προκειμένου να επωφεληθούν από την απόδοση τους καθώς η ενεργειακή χρήση αυξάνεται με την χρήση νεότερων εξοπλισμών και ικανοτήτων. Η διαδικασία αυτή είναι γνωστή και ως μετασχηματισμός του κέντρου δεδομένων. Τα πιο κοινά έργα που αφορούν τον μετασχηματισμό ενός κέντρου δεδομένων έχουν να κάνουν με την ενοποίηση, την εικονικότητα, την αυτοματοποίηση και την ασφάλεια.

- Ενοποίηση: Ο σκοπός αυτού του έργου είναι η ελάττωση του αριθμού των κέντρων δεδομένων που μπορεί να έχει ένας μεγάλος οργανισμός. Κατά αυτόν τον τρόπο μειώνεται ο αριθμός τους εξοπλισμού, του λογισμικού, των εργαλείων και διαδικασιών μέσα σε ένα κέντρο δεδομένων. Οι οργανισμοί αντικαθιστούν τον παρωχημένο εξοπλισμό με πιο εξελιγμένο που προσφέρει μεγαλύτερες ικανότητες και αποδόσεις.
- Εικονικότητα: Υπάρχει μια τάση στη χρήση IT εικονικών τεχνολογιών ώστε να αντικαθίσταται και να ενοποιείται ο πολλαπλός εξοπλισμός των κέντρων δεδομένων όπως είναι οι εξυπηρετητές. Η εικονικότητα βοηθά στη μείωση των κεφαλαιακών και λειτουργικών εξόδων των κέντρων δεδομένων και στη μείωση την κατανάλωσης ενέργειας.
- Αυτοματοποίηση: Η αυτοματοποίηση στα κέντρα δεδομένων περιλαμβάνει αυτοματοποιημένες εργασίες όπως η παροχή, η ρύθμιση, η επιδιόρθωση και η απελευθέρωση της διαχείρισης. Καθώς οι επιχειρήσεις διαθέτουν στον ενεργητικό τους λίγα άτομα με εξειδικευμένες IT ικανότητες. Οι αυτοματοποιημένες εργασίες συμβάλουν δραστικά στην αύξηση της αποδοτικότητας των κέντρων δεδομένων.
- Ασφάλεια: Στα σύγχρονα κέντρα δεδομένων, η ασφάλεια των δεδομένων στα εικονικά συστήματα είναι ενσωματωμένη με την υπάρχουσα ασφάλεια των φυσικών υποδομών. Η ασφάλεια στα σύγχρονα κέντρα

δεδομένων πρέπει να λαμβάνει υπόψιν την ασφάλεια των δικτύων και την ασφάλεια των δεδομένων των χρηστών.[10]

4.3 Προκλήσεις στα κέντρα δεδομένων

Τα συμβατικά κέντρα δεδομένων κατασκευάζονται σύμφωνα με την ιεραρχική τοπολογία δένδρου. Καθώς όμως τα κέντρα δεδομένων μεγαλώνουν σε μέγεθος πολλά προβλήματα παρατηρούνται. Παρακάτω παρουσιάζονται τα σοβαρότερα προβλήματα και οι προδιαγραφές για νέες αρχιτεκτονικές των κέντρων δεδομένων:

- Επεκτασιμότητα: Καθώς οι ανάγκες για κέντρα δεδομένων μεγαλώνουν, τα σύγχρονα κέντρα δεδομένων πρέπει να υποστηρίξουν σταδιακά επεκτασιμότητα ώστε να φιλοξενήσουν ένα τεράστιο αριθμό από εξυπηρετητές. Η κλασική αρχιτεκτονική των κέντρων δεδομένων επιτρέπει τέτοιου είδους κλιμάκωση αναβαθμίζοντας το ήδη υπάρχον υλικό (hardware) με πιο προηγμένο. Παρόλα αυτά η στρατηγική αυτή είναι ακριβή οπότε πρέπει να προταθούν αρχιτεκτονικές οι οποίες δε θα βασίζονται στην επεκτασιμότητα τους στην αντικατάσταση του παλιού υλικού (hardware).
- Στατική ανάθεση δικτύου: Στα κέντρα δεδομένων κάθε εφαρμογή αντιστοιχίζεται σε ένα εικονικό τοπικό δίκτυο (VLAN) με φυσικούς μεταγωγούς και δρομολογητές. Αυτή η απευθείας φυσική αντιστοίχιση προκαλεί στατικότητα στην ανάθεση των εξυπηρετητών. Σε ένα ιδανικό κέντρο δεδομένων κάθε εξυπηρετητής πρέπει να μπορεί να ανατεθεί σε οποιαδήποτε υπηρεσία δυναμικά σύμφωνα με τις απαιτήσεις που προκύπτουν κάθε φορά.
- Επικοινωνία μεταξύ εξυπηρετητών: Η επικοινωνία μεταξύ των εξυπηρετητών περιορίζεται από το ποσοστό υπερκάλυψης, το οποίο μπορεί να υπολογιστεί διαιρώντας το συνολικό εύρος ζώνης της σύνδεσης του εξυπηρετητή με το συγκεντρωτικό εύρος ζώνης σε επίπεδο μεταγωγέα.
- Αξιοπιστία και χρήση: Τα κλασικά κέντρα δεδομένων πάσχουν από μικρή αξιοπιστία και χρήση. Πολλαπλά μονοπάτια δεν χρησιμοποιούνται επιτυχώς στις αρχιτεκτονικές δένδρου των κέντρων δεδομένων γι' αυτό

και οι κλασσικές τοπολογίες δένδρου προσφέρουν δύο μονοπάτια ανάμεσα στους μεταγωγείς. Ένα δυναμικό μοντέλο δικτύου κέντρου δεδομένων απαιτείται έτσι ώστε να αναπτύσσονται οι πόροι του δικτύου όταν χρειάζεται να βελτιωθεί η αξιοπιστία των υπηρεσιών.

- Κόστος: Εκτιμάται ότι το κόστος σε ένα κέντρο δεδομένων με τοπολογία δένδρου είναι μια συνάρτηση του συνολικού αριθμού των τελικών υπολογιστών υποδοχής (end-hosts) με διαφορετικά ποσοστά υπερκάλυψης. Οι τελικοί μεταγωγείς που χρησιμοποιούνται από τα παραδοσιακά κέντρα δεδομένων επιβαρύνουν απαγορευτικά το κόστος για την επέκταση του μεγέθους της συστοιχίας. Το ιδανικό είναι το επεκτάσιμο εύρος ζώνης με λογικό κόστος.[12]

5. OpenStack

5.1 Τι είναι το OpenStack

Το OpenStack, είναι μια open-source πλατφόρμα cloud hosting, που ελπίζει να ανταγωνιστεί τις λύσεις cloud όπως αυτή της Microsoft και της VMware. Η RackSpace συντονίζει το project και δωρίζει τον κώδικα που κατέχει στο εγχείρημα OpenStack, στο οποίο θα ενσωματωθεί και τεχνολογία του Nebula Cloud Platform που έχει αναπτύξει η NASA.

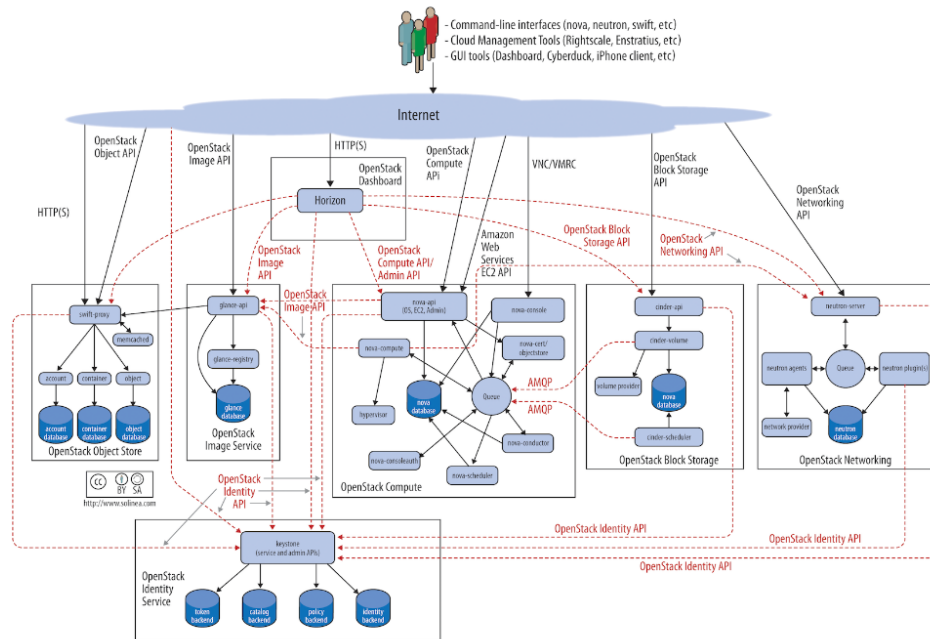
Χρησιμοποιώντας το λογισμικό OpenStack κάθε εταιρία θα μπορεί να μετατρέψει το φυσικό της hardware σε μια εσωτερικής διανομής υβριδική πλατφόρμα cloud. Η νέα αυτή open source λύση θα διανείμετε υπό μία Άδεια χρήσης Apache και θα συντηρείται από ένα μη-κερδοσκοπικού χαρακτήρα οργανισμό.

Αναφορικά με τη λειτουργία του, το OpenStack αποτελείται από ένα σύνολο ανεξάρτητων daemons ευρέως γνωστά ως OpenStack Services, καθένα από τα οποία επιτελεί μία διαφορετική λειτουργία του συστήματος. Κάθε OpenStack Service υλοποιεί τουλάχιστον από ένα Application Program Interface (API), μέσω του οποίου δέχεται αιτήματα από τελικούς χρήστες ή άλλες υπηρεσίες OpenStack.

Σχετικά με την ανταλλαγή δεδομένων, κάθε OpenStack Service επικοινωνεί με ένα διαφορετικό OpenStack Service μέσω ενός public API, με εξαίρεση περιπτώσεις που αφορά εντολές διαχειριστή συστήματος. Από την των τελικών χρηστών, η επικοινωνία με το OpenStack μπορεί να πραγματοποιηθεί με χρήση διαφόρων μεθόδων όπως:

- Μέσω του web interface που υλοποιείται από το OpenStack Dashboard
- Με χρήση ενός command line interface (CLI)
- Με χρήση εργαλείων όπως web browser plugins και του curl, μέσω των οποίων είναι δυνατή η πραγματοποίηση API calls.

Το OpenStack έχει σαν στόχο την απελευθέρωση του cloud hosting από την εξάρτηση σε ένα πάροχο, μιας και στο μέλλον οι πελάτες θα θελήσουν πλήρη ανεξαρτησία και τη δυνατότητα να μετακινούν και να επεκτείνουν την cloud υποδομή τους ανάμεσα σε διαφορετικούς παρόχους.



Εικόνα 5.1: Αναπαράσταση λειτουργίας OpenStack

5.2 Αρχιτεκτονική του OpenStack

Το OpenStack αποτελείται από Projects που το καθένα διαχειρίζεται μια υπηρεσία:

- **Horizon (Dashboard)**

Παρέχει ένα web-based portal που αλληλοεπιδρά με τις υποκείμενες υπηρεσίες του Openstack (Nova, Neutron, Cinder κλπ) μέσω ενός συνόλου από GUIs.

Το σύστημα Horizon, αναλύεται πρακτικά σε τρεις (3) βασικούς πίνακες ελέγχου:

- Τον πίνακα ελέγχου χρήστη (User Dashboard),
- Τον πίνακα ελέγχου του συστήματος OpenStack (System Dashboard),
- Τον πίνακα ελέγχου ρυθμίσεων (Settings Dashboard)

- **Nova (Compute)**

Διαχειρίζεται τα instances σε ένα περιβάλλον OpenStack. Είναι υπεύθυνο για την κατά απαίτηση δημιουργία, προγραμματισμό και παροπλισμό των εικονικών μηχανών.

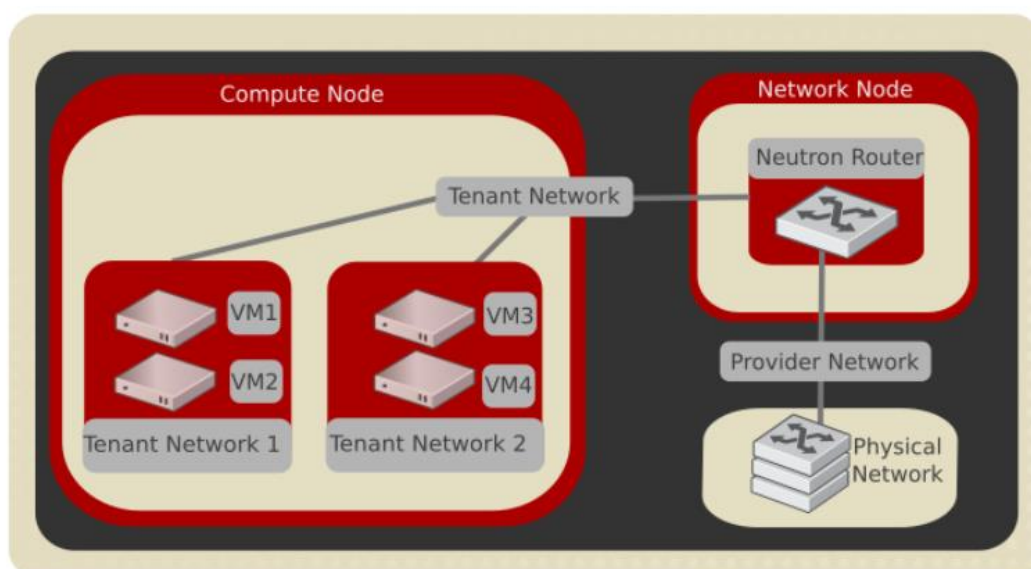
Τυπικά, η υπηρεσία Nova επικοινωνεί με την υπηρεσία Glance για την χρήση των κατάλληλων εικόνων συστήματος καθώς και με την υπηρεσία Cinder για

δέσμευση και προσάρτηση δίσκων μόνιμης αποθήκευσης προς τα εικονικά συστήματα.

- **Neutron (Networking)**

Ενεργοποιεί την υπηρεσία της σύνδεσης δικτύου για τις άλλες υπηρεσίες του OpenStack. Παρέχει ένα API στους χρήστες για τον ορισμό δικτύων και την προσάρτηση σε αυτά.

Αρχικά, ο host που είναι υπεύθυνος για την παροχή των δικτυακών λειτουργιών, χρησιμοποιεί τον neutron-server daemon, μέσω του οποίου παρέχει ένα networking API για την διαχείριση του εκάστοτε networking plug-in. Το τελευταίο παρέχει τα απαραίτητα L2 capabilities του συστήματος ενώ ταυτόχρονα δρα ως baseline για την προσάρτηση third-party τεχνολογιών δικτύωσης.



Εικόνα 5.2: OpenStack Neutron

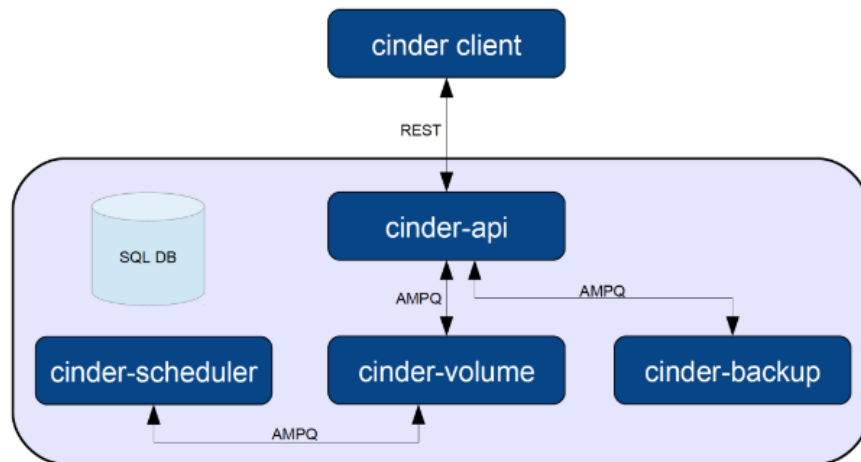
- **Swift (Object Storage)**

Το Swift, προσφέρει την υπηρεσία αποθήκευσης δεδομένων μεταβλητού μεγέθους υπό τη μορφή αντικειμένων (όπως π.χ. φωτογραφίες, video και αρχεία κειμένου).

- **Cinder (Block Storage)**

Παρέχει την αποθήκευση για block αρχεία που χρησιμοποιούνται για την λειτουργία των instances. Αυτοί τυπικά, προσαρτώνται σε instances εικονικών συστημάτων, διαμέσου επικοινωνίας της υπηρεσίας Cinder με την υπηρεσία Nova,

ως συσκευές μόνιμης αποθήκευσης. Μέσω αυτού γίνεται η δημιουργία και η διαχείριση συσκευών block storage.



Εικόνα 5.3: OpenStack Cinder

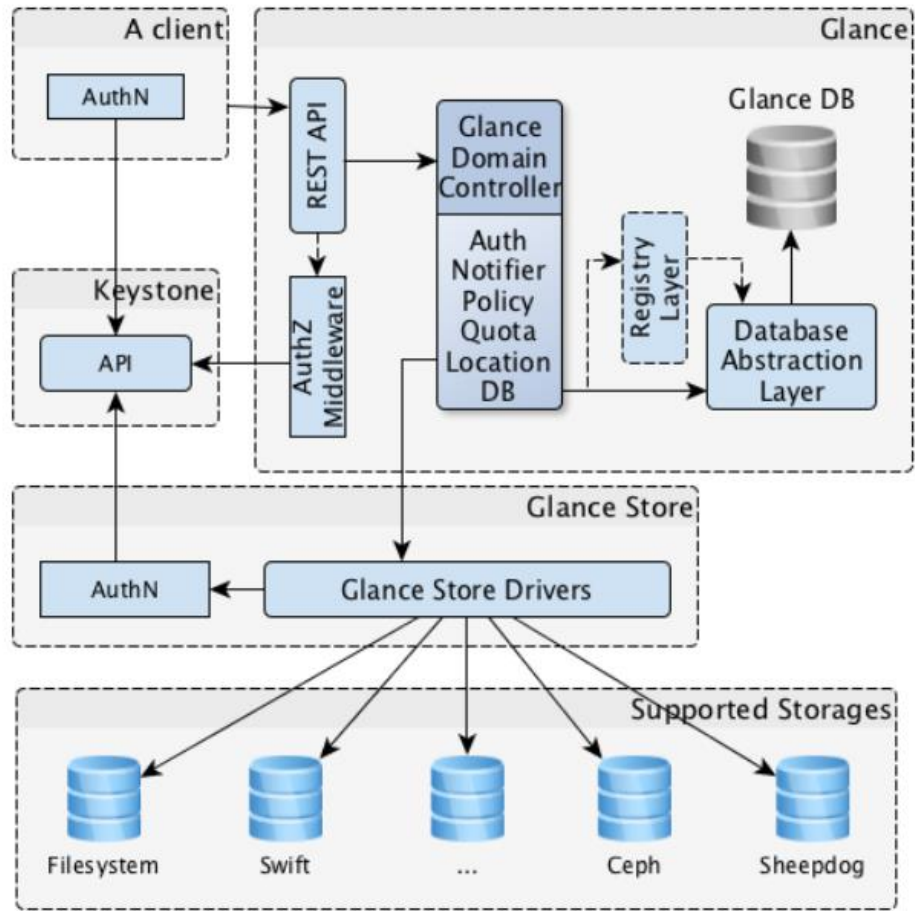
- **Keystone (Identity Service)**

Αποτελεί μία από τις σημαντικότερες υπηρεσίες του OpenStack καθώς, αποτελεί τον μηχανισμό ταυτοποίησης μέσω του οποίου οι τελικοί χρήστες, αφενός πιστοποιούνται προς το σύστημα και αφετέρου, ενημερώνονται για το σύνολο των πόρων και υπηρεσιών που έχουν δικαίωμα να χρησιμοποιήσουν.

- **Glance (Image Service)**

Η υπηρεσία αυτή, προσφέρει τη δυνατότητα ανεβάσματος (upload) και εύρεσης στοιχείων δεδομένων (που διαμοιράζονται και με άλλες υπηρεσίες), όπως εικόνες λειτουργικών συστημάτων (images) καθώς και metadata definitions για αυτές.

Η βασική αρχιτεκτονική της υπηρεσίας Glance περιγράφεται στην παρακάτω εικόνα:



Εικόνα 5.4: OpenStack Cinder

- **Ceilometer (Telemetry)**

Επιτηρεί και μετρά το OpenStack για benchmarking, επεκτασιμότητα, στατιστικούς σκοπούς και χρεώσεις.

5.2.1 Keystone

Το Keystone είναι η υπηρεσία που είναι υπεύθυνη για την πιστοποίηση των χρηστών και των υπηρεσιών.

sudo apt-get install keystone python-keystone python-keystoneclient

Ανοίγουμε το αρχείο `/etc/keystone/keystone.conf` και αλλάζουμε την γραμμή

admin_token = ADMIN

Να εμφανίζεται:

admin_token = admin

Εφόσον η βάση θα αποθηκεύει τις ρυθμίσεις keystone, αντικαθιστούμε την ακόλουθη γραμμή στο αρχείο:

```
connection = sqlite:///var/lib/keystone/keystone.db
```

με την παρακάτω:

```
connection = mysql://keystonedbadmin:keystonesecret@10.10.10.2/keystone
```

Επανεκκίνηση του keystone:

```
sudo service keystone restart
```

Μετά ακολουθεί ο συγχρονισμός της βάσης:

```
sudo keystone-manage db_sync
```

Δημιουργία Χρηστών

Η παρακάτω εντολή δημιουργεί έναν χρήστη για την σύνδεση στο dashboard:

```
keystone user-create --name admin --pass admin --email admin@admin.gr
```

Δημιουργία Roles

Με την παρακάτω εντολή επιτυγχάνεται η δημιουργία ενός ρόλου:

```
keystone role-create --name admin
```

Προσθήκη ρόλων

Η προσθήκη ρόλων στους χρήστες γίνεται με παρακάτω εντολή:

```
keystone user-role-add --user $USER_ID --role $ROLE_ID --tenant_id $TENANT_ID
```

5.2.2 Glance

Εγκατάσταση του Glance:

```
sudo apt-get install glance-api glance-client glance-common glance-registry  
python-glance
```

Κάνουμε τις ακόλουθες αλλαγές στο αρχείο /etc/glance/glance-api-paste.ini από:

```
admin_tenant_name = %SERVICE_TENANT_NAME%
```

```
admin_user = %SERVICE_USER%
```

```
admin_password = %SERVICE_PASSWORD%
```

σε:

```
admin_tenant_name = glance
```

admin_user = admin

admin_password = admin

Επίσης κάνουμε παρόμοιες αλλαγές στο /etc/glance/glance-registry-paste.ini:

[paste_deploy]

flavor = keystone

Τέλος τρέχουμε την παρακάτω εντολή για να κάνουμε συγχρονισμό με την βάση:

glance-manage db_sync

Κάνουμε restart το glance-api για ενημερωθούν οι αλλαγές που έχουμε κάνει:

sudo restart glance-api

sudo restart glance-registry

5.2.3 Nova

Επεξεργασία αρχείου /etc/nova/nova.conf με τα παρακάτω:

--dhcpbridge_flagfile=/etc/nova/nova.conf

--dhcpbridge=/usr/bin/nova-dhcpbridge

--allow_admin_api=true

--use_deprecated_auth=false

--auth_strategy=keystone

--s3_host=10.10.10.2

--ec2_host=10.10.10.2

--cc_host=10.10.10.2

--libvirt_type=kvm

--libvirt_use_virtio_for_bridges=true

--start_guests_on_host_boot=true

--novnc_enabled=true

--public_interface=eth0

--flat_interface=eth1

```
--iscsi_helper=tgtadm
```

```
--root_helper=sudo nova-rootwrap
```

```
--verbose
```

Μετά αλλάζουμε τα δικαιώματα στον φάκελο και στον αρχείο:

```
sudo chown -R nova:nova /etc/nova
```

```
sudo chmod 644 /etc/nova/nova.conf
```

Τέλος κάνουμε την συγκεκριμένη αλλαγή στο αρχείο /etc/nova/api-paste.ini

```
admin_tenant_name = %SERVICE_TENANT_NAME%
```

```
admin_user = %SERVICE_USER%
```

```
admin_password = %SERVICE_PASSWORD%
```

```
-----
```

```
admin_tenant_name = glance
```

```
admin_user = admin
```

```
admin_password = admin
```

Μετά από τις παραπάνω αλλαγές πρέπει να κάνουμε restart τα παρακάτω services:

```
sudo restart nova-network
```

```
sudo restart nova-xcompute
```

```
sudo restart nova-api
```

```
sudo restart nova-volume
```

5.2.4 OpenStack Dashboard

Η εγκατάσταση γίνεται με το εξής command:

```
sudo apt-get install openstack-dashboard
```

Και γίνεται η επανεκκίνηση του Apache

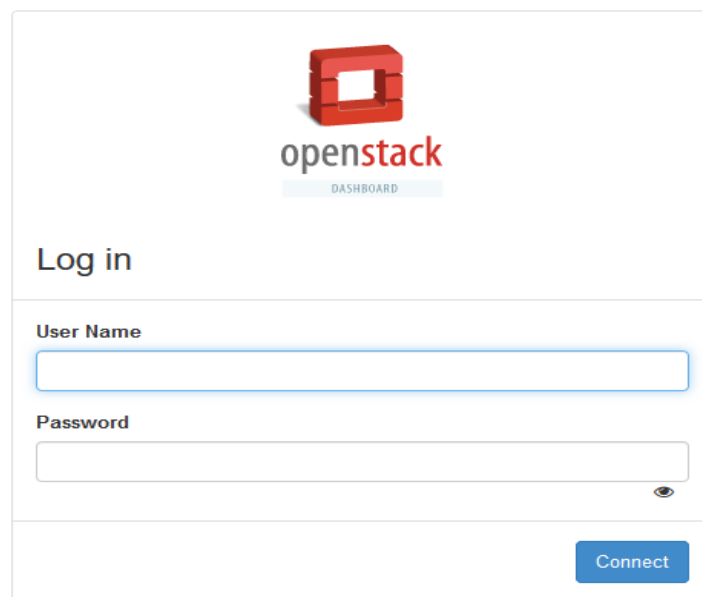
```
service apache2 restart
```

Μόλις γίνουν οι παραπάνω παραμετροποιήσεις εκτελούμε την τελική εντολή

```
packstack --allinone
```

για να γίνει η σύνδεση όλων των παραπάνω services και να μπορέσουμε να συνδεθούμε στο OpenStack μέσω του browser. Η σύνδεση γίνεται αν βάλουμε την IP μας στον Browser, στην περίπτωση μας είναι η 192.168.1.53, και μπορούμε αποκτήσουμε πρόσβαση σε όλα τα services μέσω web interface.

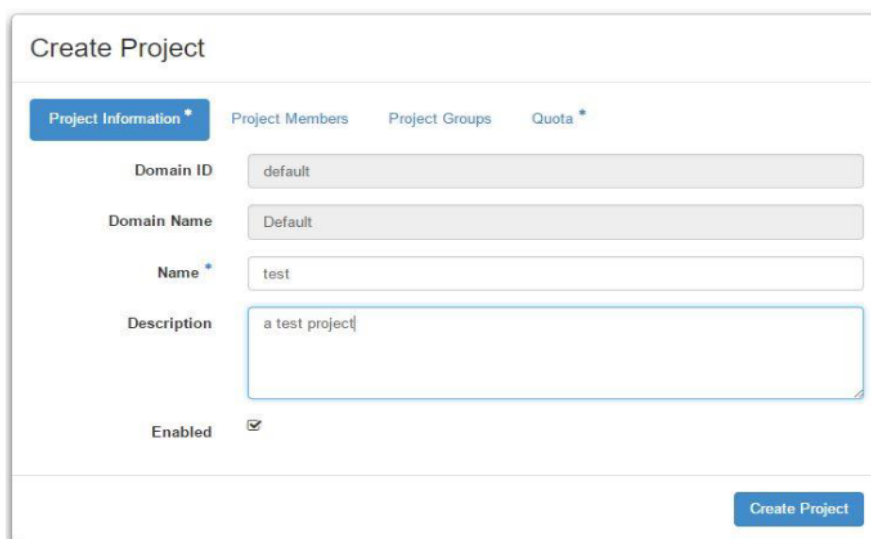
5.3 OpenStack Dashboard (Προεπισκόπηση) Είσοδος στο Dashboard (<http://192.168.1.53>)



The screenshot shows the OpenStack Dashboard login interface. At the top center is the OpenStack logo, which consists of a red 3D cube with a square hole in the center, and the text "openstack" below it, with "DASHBOARD" in smaller letters underneath. Below the logo is the heading "Log in". There are two input fields: "User Name" and "Password". The "Password" field has a small eye icon to its right for toggling visibility. At the bottom right of the form is a blue button labeled "Connect".

Εικόνα 5.5: Log in OpenStack Dashboard

Δημιουργία Project



The screenshot shows the "Create Project" form in the OpenStack Dashboard. The form has a title "Create Project" at the top. Below the title are four tabs: "Project Information" (which is active and highlighted in blue), "Project Members", "Project Groups", and "Quota". The "Project Information" tab contains several fields: "Domain ID" with the value "default", "Domain Name" with the value "Default", "Name" with the value "test", and "Description" with the value "a test project". There is also a checkbox labeled "Enabled" which is checked. At the bottom right of the form is a blue button labeled "Create Project".

Εικόνα 5.6: Project creation

Project Overview

Εικόνα 5.7: Project Overview

OpenStack Nova

Εικόνα 5.8: Instance Details

Επιλέγουμε τον αποθηκευτικό χώρο για το Instance.

Name	Description	Size	Type	Availability Zone
> 90395b12-9d4c-4c2a-99c3-e1fdb54e0d39		1 GB	qcow2	nova
> 48ba07a0-dcd0-49ae-93f6-5626e6a22285		1 GB	qcow2	nova
> 29613f1e-4214-4556-92a2-5e1f8d352cd4		1 GB	qcow2	nova

Εικόνα 5.9: Instance Source

Επιλέγουμε τον τύπο του Instance που θέλουμε να χρησιμοποιήσουμε.

The screenshot shows the 'Launch Instance' dialog box with the 'Flavor' tab selected. The 'Available' section contains a table of instance flavors:

Name	VCPUS	RAM	Total Disk	Root Disk	Ephemeral Disk	Public
m1.tiny	1	512 MB	1 GB	1 GB	0 GB	Yes
m1.small	1	2 GB	20 GB	20 GB	0 GB	Yes
m1.medium	2	4 GB	40 GB	40 GB	0 GB	Yes
m1.large	4	8 GB	80 GB	80 GB	0 GB	Yes
m1.xlarge	8	16 GB	160 GB	160 GB	0 GB	Yes

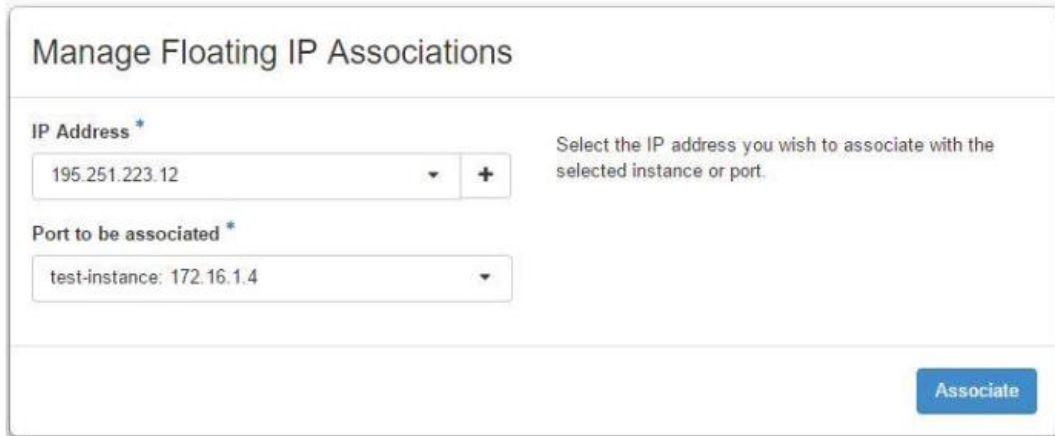
Εικόνα 5.10: Instance Hardware Details

Ανάθεση διεύθυνσης floating-ip στο instance

Η διεύθυνση floating ip, αποτελεί πρακτικά την διεύθυνση δικτύου μέσω της οποίας, ένα instance γίνεται διαθέσιμο προς προσπέλαση από εξωτερικά δημόσια δίκτυα.

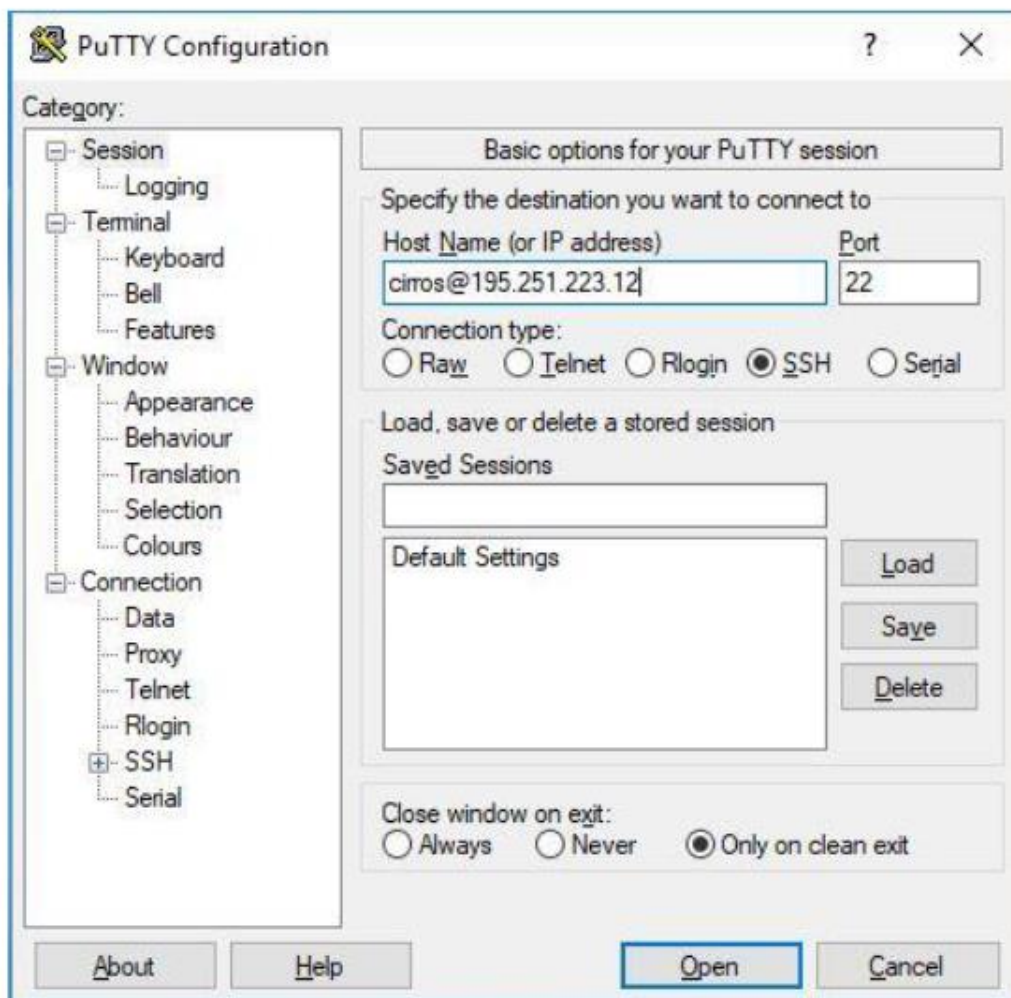
The screenshot shows the 'Manage Floating IP Associations' dialog box. It includes a dropdown for 'IP Address' currently showing 'No floating IP addresses allocated' and a dropdown for 'Port to be associated' showing 'test-instance: 172.16.1.4'. An 'Associate' button is located at the bottom right.

Εικόνα 5.11: Floating IP



Εικόνα 5.12: Floating IP (2)

Σύνδεση SSH στο instance



Εικόνα 5.13: Remote Connection with Putty


```

195.251.223.12 - PuTTY
Using username "cirros".
cirros@195.251.223.12's password:
$
$ ping -c 4 www.openstack.org
PING www.openstack.org (104.20.110.33): 56 data bytes
64 bytes from 195.251.223.12: seq=0 ttl=53 time=166.896 ms
64 bytes from 195.251.223.12: seq=1 ttl=53 time=168.206 ms
64 bytes from 195.251.223.12: seq=2 ttl=53 time=169.008 ms
64 bytes from 195.251.223.12: seq=3 ttl=53 time=168.562 ms

--- www.openstack.org ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 166.896/168.168/169.008 ms
$ _

```

Εικόνα 5.14: Remote Connection

OpenStack Glance

Ανεβάζουμε τα Images που θέλουμε να χρησιμοποιήσουμε.

The screenshot shows the OpenStack Glance 'Images' page. The interface includes a sidebar with navigation options like 'Project', 'Admin', and 'System'. The main content area displays a table of images with columns for Image Name, Type, Status, Public, Protected, Format, Size, and Actions. There are also buttons for 'Create Image' and 'Delete Images'.

Image Name	Type	Status	Public	Protected	Format	Size	Actions
Ubuntu 14.04 - Cloud Image	Image	Active	Yes	No	QCOW2	2.2 GB	Edit Image
Fedora 23 Cloud Atomic Host	Image	Queued	Yes	No	QCOW2	0 bytes	Edit Image
Fedora 23 Cloud	Image	Queued	Yes	No	QCOW2	0 bytes	Edit Image
CentOS 7 x86_64	Image	Active	Yes	No	QCOW2	8.0 GB	Edit Image

Εικόνα 5.15: List of Uploaded images

Συμπληρώνουμε τα στοιχεία του Image και ανεβάζουμε το αρχείο .iso στο OpenStack.

Create An Image ✕

Name *

Description

Image Source

Image Location ⓘ

Format *

Architecture

Minimum Disk (GB) ⓘ

Minimum RAM (MB) ⓘ

Copy Data ⓘ

Public

Protected

Description:

Specify an image to upload to the Image Service.

Currently only images available via an HTTP URL are supported. The image location must be accessible to the Image Service. Compressed image binaries are supported (.zip and .tar.gz.)

Please note: The Image Location field MUST be a valid and direct URL to the image binary. URLs that redirect or serve error pages will result in unusable images.

Εικόνα 5.16: Image Creation (Glance)

OpenStack Cinder

Παρουσιάζει τους αποθηκευτικούς πόρους στους χρήστες που μπορούν να καταναλωθούν από το Openstack Compute Project (Nova).

Volumes		Volume Snapshots								
Name	Description	Size	Status	Type	Attached To	Availability Zone	Bootable	Encrypted	Actions	
<input type="checkbox"/>	sdnc-volume	-	100GB	Available	-		nova	No	No	Edit Volume

Displaying 1 item

Εικόνα 5.17: Αποθηκευτικός χώρος (Cinder)

Δημιουργούμε Volumes για αποθήκευση ανάλογα με τις ανάγκες των χρηστών.

Create Volume

×

Volume Name: *

Description:

Type:

Size (GB): *

Volume Source:

Availability Zone

Description:

Volumes are block devices that can be attached to instances.

Volume Limits

Total Gigabytes (0 GB) 1,000 GB Available

Number of Volumes (0) 10 Available

Cancel

Create Volume

Εικόνα 5.18: Volume Creation

Keystone

Εδώ δημιουργούμε τους χρήστες που θα έχουν πρόσβαση στο σύστημα.

The screenshot displays the OpenStack Keystone interface. At the top left, the OpenStack logo is visible. The user 'admin' is logged in, as indicated by the 'admin' dropdown menu. In the top right corner, the user 'chris' is shown with a 'Sign Out' link. The main content area is titled 'Users' and contains a table of existing users. A red arrow points to the '+ Create User' button in the top right of the table area. Another red arrow points to the 'Users' link in the left sidebar navigation menu.

<input type="checkbox"/>	User Name	Email	User ID	Enabled	Actions
<input type="checkbox"/>	heat	heat@localhost	0da46615399046279fca25e94ad08b79	True	Edit ▾
<input type="checkbox"/>	geoff	geoff@starkandwayne.com	135878c4a8c4452180356b21208e911a	True	Edit ▾
<input type="checkbox"/>	nova	nova@localhost	3e83b0610091408bb797271779a64723	True	Edit ▾
<input type="checkbox"/>	chris		602666ebd8904f08a5c7e7644260ec0c	True	Edit ▾
<input type="checkbox"/>	admin	admin@localhost	6eacdaf4cabd48029489afa2c986f6c5	True	Edit ▾

Εικόνα 5.19: Λίστα Χρηστών

Φτιάχνουμε security groups και με τα ανάλογα δικαιώματα για τον κάθε χρήστη, διότι δεν πρέπει όλοι οι χρήστες να έχουν admin δικαιώματα.

Project / Network / Security Groups

Security Groups

ben **3** **4** + Create Security Group Delete Security Groups

Displaying 2 items

<input type="checkbox"/>	Name	Security Group ID	Description	Actions
<input type="checkbox"/>	ben-again	7f9004ed-478f-4ef1-aba1-28f6f0570059	test	Manage Rules ▾
<input type="checkbox"/>	ben-test	6ca8a772-3a54-4bb9-9b4e-6f4b63f97d07		Manage Rules ▾

Displaying 2 items

Εικόνα 5.20: Security Groups

OpenStack Neutron

Από εδώ διαχειριζόμαστε το δίκτυο του συστήματος.

Create Network

Network Subnet Subnet Details

Network Name
private-network

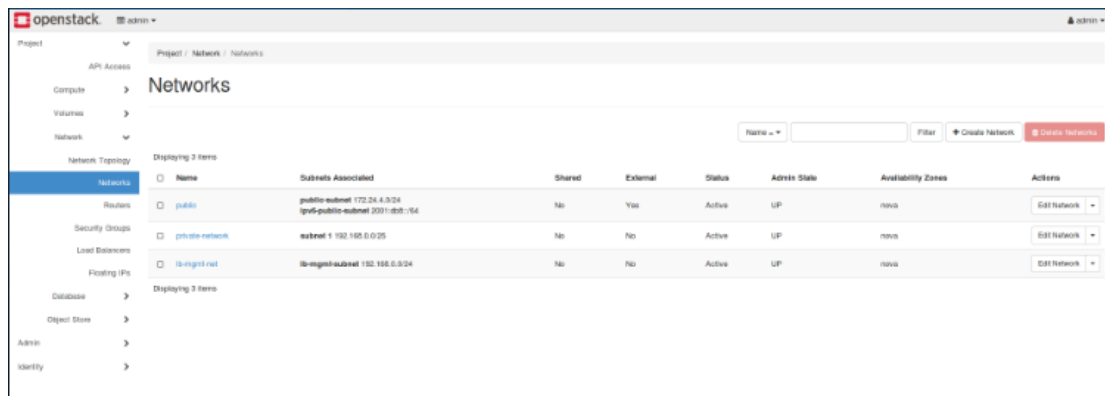
Create a new network. In addition, a subnet associated with the network can be created in the following steps of this wizard.

Enable Admin State ⓘ
 Shared
 Create Subnet

Availability Zone Hints ⓘ
nova

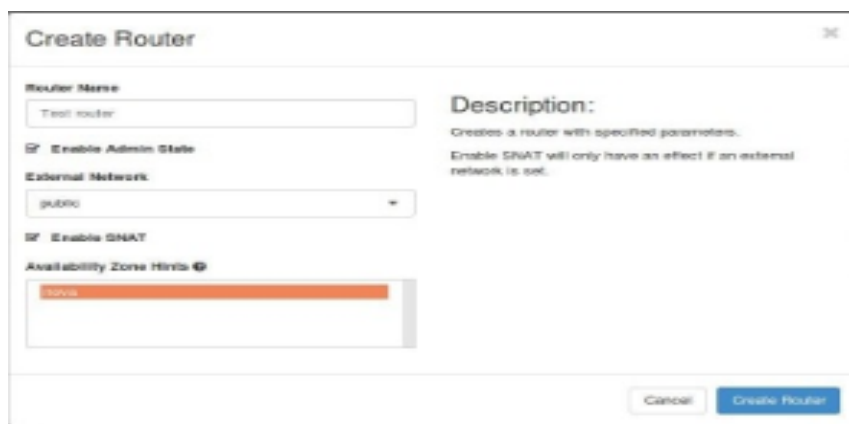
Cancel Back Next

Εικόνα 5.21: Network Creation



Εικόνα 5.22: Network List

Και το Router που συνδέει τα δίκτυα μεταξύ τους.



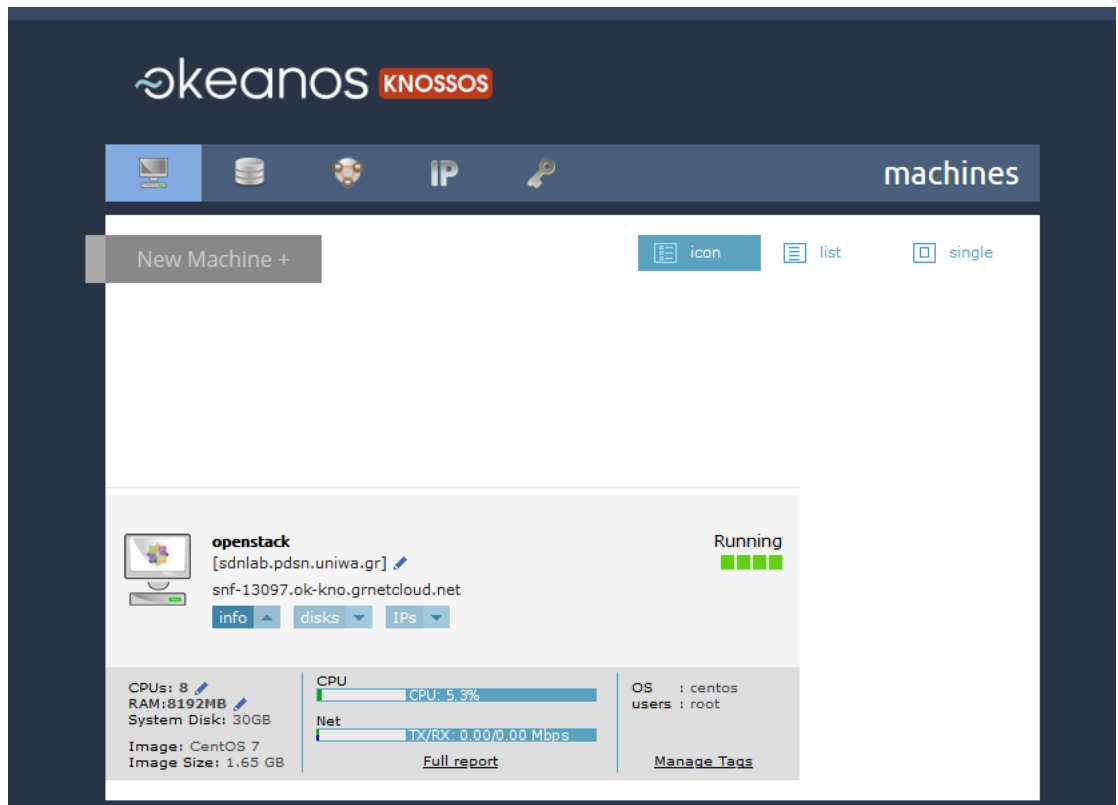
Εικόνα 5.23: Virtual Router Creation

Υλοποίηση και συμπεράσματα

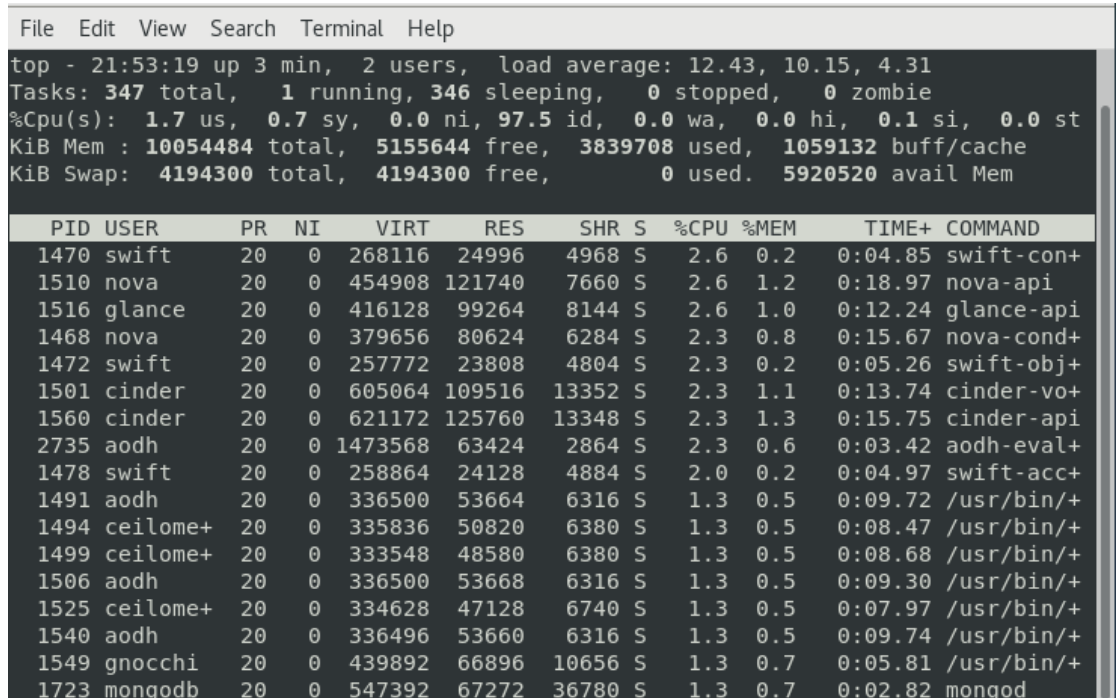
Η υλοποίηση έγινε πάνω στην πλατφόρμα του ~Οκεανος η οποία είναι μια πλατφόρμα IaaS, όπως είναι και το OpenStack.

Για την εγκατάσταση του OpenStack χρησιμοποιήθηκε το RDO, το οποίο δίνει την δυνατότητα να στηθεί το OpenStack με την χρήση απλών script.

Το εικονικό μηχάνημα που δημιουργήσαμε στον Ωκεανό έχει Centos 8 x64, 8GB RAM και επεξεργαστή με 8 πυρήνες. Η αρχιτεκτονική του μηχανήματος πρέπει να είναι πάντα x64 για να μπορέσει να λειτουργήσει το OpenStack.



Εικόνα 5.24: VM Resources Info



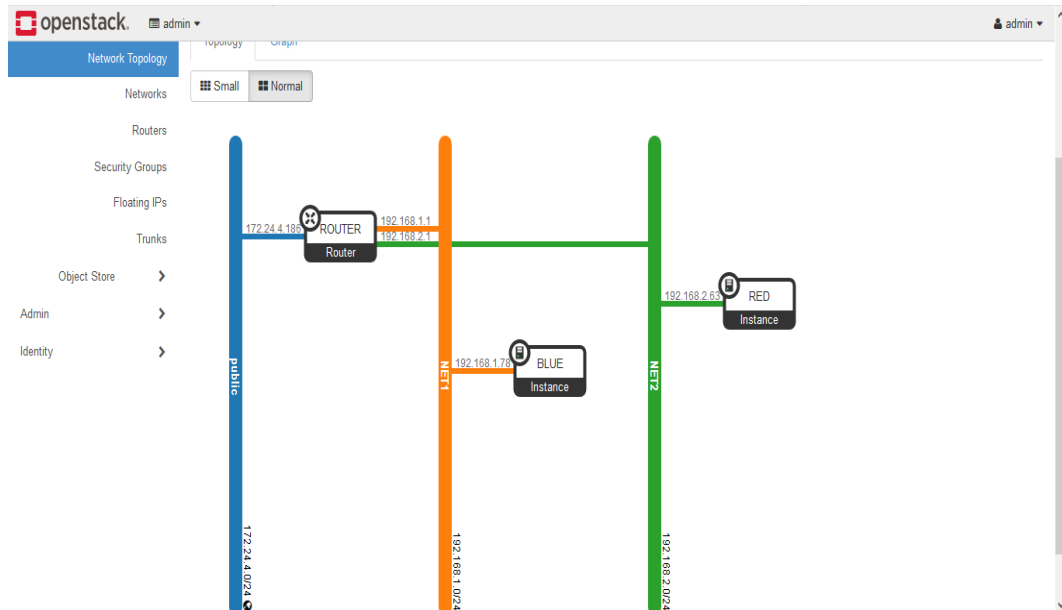
Εικόνα 5.25: OpenStack Services

Η υλοποίηση της εργασίας έγινε με την χρήση των εντολών καθώς και των παραμετροποιήσεων των αρχείων που αναφέρθηκαν στον κεφάλαιο 5.2 . Στην

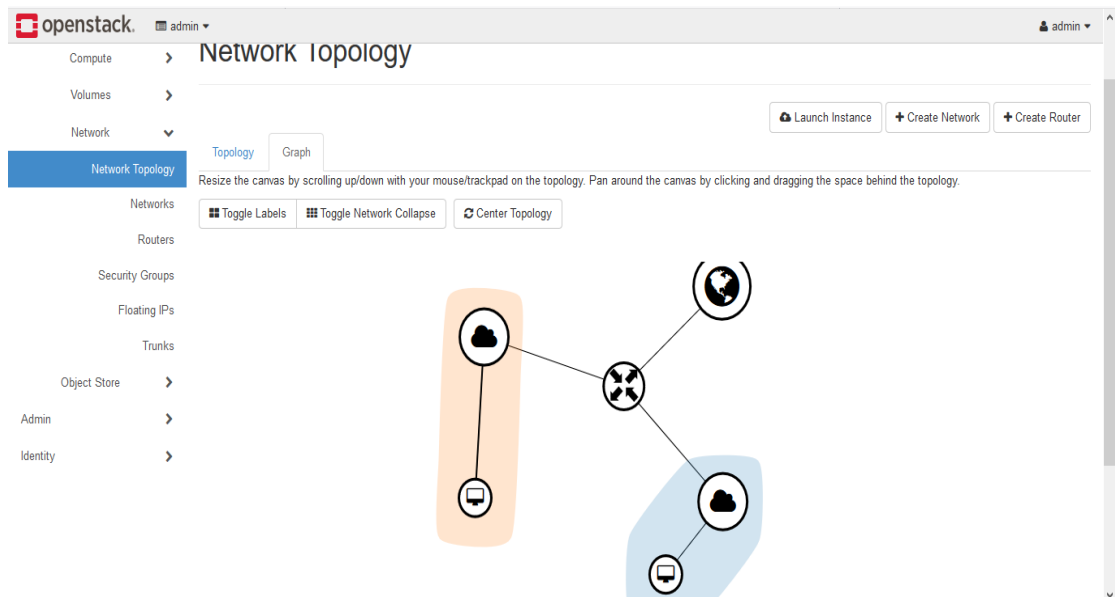
συνέχεια χρησιμοποιήθηκε η παρακάτω εντολή για την σύνδεση όλων των υπηρεσιών μεταξύ τους:

packstack –allinone

Το τελικό σύστημα που υλοποιήθηκε βρίσκεται στις παρακάτω εικόνες. Περιέχει δύο Virtual Machines τα οποία βρίσκονται σε διαφορετικά δίκτυα, ένα virtual router το οποίο συνδέει τα δύο αυτά δίκτυα καθώς και με το internet και δύο DHCP Server οι οποίοι διαμοιράζουν IP's στα virtual machines.



Εικόνα 5.25: Απεικόνιση Δικτύου



Εικόνα 5.26: Απεικόνιση Δικτύου (2)

Από τα αποτελέσματα της εργασίας, προκύπτει ότι η χρήση του λογισμικού OpenStack, μπορεί να προσφέρει συστήματα υποδομής νέφους, τα οποία να ικανοποιούν τα κριτήρια μιας μικρής ή μεγάλης κλίμακας εταιρείας. Επιπρόσθετα, η χρήση συστημάτων αυτοματισμού, προσέφερε ένα γρήγορο και αποτελεσματικό τρόπο διαμόρφωσης του περιβάλλοντος OpenStack, συγκριτικά με παραδοσιακές μεθόδους χειροκίνητης διαμόρφωσης υπηρεσιών από τον ίδιο τον χρήστη.

Δεδομένου ότι για την υλοποίηση της εφαρμογής έγινε χειροκίνητη εγκατάσταση κάθε υπηρεσίας σε περιβάλλον Centos 8 παρατηρήθηκαν μερικά σφάλματα τα οποία στην συνέχεια διορθώθηκαν. Επίσης η αυξημένες ανάγκες σε υπολογιστικούς πόρους του λογισμικού OpenStack, είχε ως αποτέλεσμα πολλές φορές να τερματιστεί απροσδόκητα η λειτουργία του Virtual Machine που φιλοξενούσε το λογισμικό.

Μελλοντικές αναβαθμίσεις

Όσον αφορά τη διαδικασία ανάπτυξης του συστήματος, μία πιθανή επέκταση αποτελεί η δημιουργία επιπλέον compute nodes, ώστε με την επιπρόσθετη υπολογιστική ισχύ, να δοθεί η δυνατότητα φιλοξενίας μεγαλύτερου αριθμού εικονικών μηχανών.

Επιπρόσθετα, σαν μία ακόμα επέκταση συστήματος, μπορούν να προστεθούν και νέες υπηρεσίες, ώστε να είναι σε θέση να προσφέρει περισσότερη λειτουργικότητα στο χρήστη. Το OpenStack προσφέρει αρκετές νέες υπηρεσίες που μπορούν να προστεθούν για την υλοποίηση project ανάλογα με τις απαιτήσεις της εταιρείας όπως:

- Το **Ironic** το οποίο προσφέρει τη δυνατότητα δημιουργίας νέων instances πάνω σε μία φυσική υπολογιστική οντότητα (bare metal server), έναντι της χρήσης εικονικών μηχανών.
- Το **Manila**, για την παροχή κοινόχρηστου χώρου αποθήκευσης μεταξύ των instances (σε αντίθεση π.χ. με το block storage το οποίο προσφέρει η υπηρεσία Cinder).
- Το **Trove**, μέσω του οποίου οι χρήστες έχουν τη δυνατότητα να διαμορφώνουν και να διαχειρίζονται βάσεις δεδομένων (Database as a Service).
- Το **Sahara** που παρέχει τη δυνατότητα ενσωμάτωσης data processing frameworks (όπως π.χ. Hadoop, Spark και Storm τα οποία χρησιμοποιούνται στον κλάδο του big data analysis) στο περιβάλλον του OpenStack.

- Το **RabbitMQ** το οποίο είναι ένας μεσολαβητής ανταλλαγής μηνυμάτων. Παρέχει στην εφαρμογή μια κοινή πλατφόρμα για την αποστολή και λήψη μηνυμάτων καθώς και την αποθήκευση τους σε ένα ασφαλές μέρος.

Τέλος, επεκτείνοντας τη χρήση των containers και των πλεονεκτημάτων που αυτά προσφέρουν, μία ακόμα μελλοντική επέκταση, θα μπορούσε να αποτελέσει η χρήση του Docker hypervisor driver στον compute node του συστήματος. Μέσω του τελευταίου, δίνεται η δυνατότητα της εκτέλεσης εφαρμογών ενός instance σε application containers, προσφέροντας έτσι εφαρμογές πλήρως αποδεσμευμένες από το software και το hardware του εκάστοτε host.

Βιβλιογραφία

- [1] ONF White Paper, “Software-Defined Networking: The New Norm for Networks” ,April 2012
- [2] Open Networking Foundation (ONF), “SDN Architecture Overview” ,Version 1.0, December 2013
- [3] OpenFlow Specifications Sheet v1.3.3 2013.
- [4] "What is an OpenFlow Controller?",
<https://www.sdxcentral.com/networking/sdn/definitions/openflow-controller/>
- [5] "OpenFlow Switch: What Is It and How Does it Work?", <http://www.cablesolutions.com/whats-openflow-switch-how-it-works.html>
- [6] “OpenDaylight”, <https://www.opendaylight.org>
- [7] “What Is RESTCONF”, <http://sdntutorials.com/what-is-restconf/>
- [8] “Mininet Overview“, <http://mininet.org/overview/>
- [9] “Introduction to Mininet“, <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
- [10] <http://www.ieee802.org/11/>
- [12] Andrew S. Tanenbaum,“Computer Networks”, Fourth Edition
- [11] ”Data center”, http://en.wikipedia.org/wiki/Data_center