ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: «Images εργαστηρίων ως cloud υπηρεσία»

Κωνσταντινίδης Κωνσταντίνος(07/3211)

kkonstan@it.teithe.gr

Σωτηρίου Γεώργιος(06/3104)

geosot@it.teithe.gr

Επιβλέπων καθηγητής: Δρ. Ηλιούδης Χρήστος

iliou@it.teithe.gr

Θεσσαλονίκη 2015

ΠΡΟΛΟΓΟΣ

Στον κύκλο τον ανθρώπων της τεχνολογίας και όσων απαρτίζουν την βιομηχανία της Πληροφορικής, ένα από τα κύρια και πιο συχνά θέματα συζήτησης αφορά τα υπολογιστικά νέφη. Οι ειδικοί πάνω στο τομέα αυτόν κάνουν λόγο για μια τεχνολογία που μπορεί να αλλάξει ριζικά τα πράγματα γύρω από την επιστήμη των υπολογιστών, όπως την ξέρουμε μέχρι σήμερα. Κολοσσιαίοι οργανισμοί και επιχειρήσεις έχουν ρίξει την προσοχή τους στην ανάπτυξη υπολογιστικών νεφών και στην μεταφορά των ήδη υπαρχόντων συστημάτων τους σε αυτά.

Ο κλάδος της εκπαίδευσης δεν θα μπορούσε να απέχει και να μην επηρεαστεί από αυτήν την τεχνολογική καινοτομία. Η ανάπτυξη ενός συστήματος υπολογιστικού νέφους, που θα εξυπηρετεί της ανάγκες ενός εκπαιδευτικού ιδρύματος, μπορεί να έχει πολλά οφέλη για αυτό και για τους φοιτητές του. Ένα τέτοιο σύστημα προσπαθήσαμε να αναπτύξουμε και εμείς με σκοπό να επωφεληθεί, στο μέλλον, το ίδρυμα μας από αυτό.

ΠΕΡΙΛΗΨΗ

Το ερευνητικό τμήμα αυτής της πτυχιακής εργασίας είναι η ανάπτυξη και η δημιουργία ενός συστήματος υπολογιστικού νέφους με σκοπό τον διαμοιρασμό εικόνων λειτουργικών συστημάτων προς τους φοιτητές του τμήματος Μηχανικών Πληροφορικής του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης. Η ανάπτυξη ενός τέτοιου περιβάλλοντος, αποτελεί μια πολύ ενδιαφέρουσα προοπτική με εφαρμογές που θα διευκόλυναν την εκπαιδευτική διαδικασία του φοιτητή, το εκπαιδευτικού προσωπικού και του ιδρύματος γενικότερα.

Συγκεκριμένα, ο φοιτητής θα μπορούσε με την χρήση ενός τέτοιου συστήματος να έχει το προσωπικό λειτουργικό περιβάλλον του σε οποιονδήποτε χώρο επιθυμούσε (είτε στο εσωτερικό δίκτυο του ιδρύματος είτε εξωτερικά με την χρήση VPN). Επίσης θα μπορούσε να έχει πρόσβαση στο σύστημα οποιαδήποτε στιγμή της ημέρας. Επίσης, το εκπαιδευτικό προσωπικό θα μπορεί να επωφελείται από την χρήση ενός τέτοιου συστήματος καθώς θα μπορεί να διαμορφώνει ξεχωριστές εικόνες λειτουργικών συστημάτων με προγράμματα και υπηρεσίες που επιθυμεί. Το παραπάνω χαρακτηριστικό θα διευκόλυνε κατά πολύ την εκπαιδευτική διαδικασία εν ώρα μαθήματος και θα εμπλούτιζε κατά πολύ το εκπαιδευτικό υλικό του εκάστοτε μαθήματος. Τέλος, το ίδιο το ίδρυμα θα είχε οφέλη από την δημιουργία ενός τέτοιου συστήματος τόσο σε οικονομικό όσο και σε άλλα επίπεδα. Ας σκεφτούμε το γεγονός, πώς το ίδρυμα θα πρέπει να συντηρεί ένα μηχάνημα εν αντιθέσει με τα εκατοντάδες που υπάρχουν στην παρούσα φάση. Επίσης θα μπορούσε να αυξηθεί η εν δυνάμει χωρητικότητα των εργαστηριακών τμημάτων καθώς θα ήταν ανεξάρτητη από τον περιορισμένο αριθμό προσωπικών υπολογιστών.

Τελικά, το σύστημα νέφους που δημιουργήσαμε, μας δίνει την δυνατότητα σύνδεσης σε αυτό μέσω ενός απλού φυλλομετρητή. Στην συνέχεια μέσω του λογισμικού μας δίνεται η δυνατότητα χρήσης και διαχείρισης του συστήματος με σκοπό την δημιουργία εικόνων προς διαμοιρασμό. Επίσης, το σύστημα μας δίνει την δυνατότητα χρήσης και άλλων πολλών εργαλείων προσφέροντας μια ολοκληρωμένη εμπειρία χρήσης ενός υπολογιστικού νέφους.

Οι τεχνολογίες που χρησιμοποιήθηκαν ήταν το λογισμικό νέφους ανοιχτού κώδικα Openstack. Επίσης, το λειτουργικό σύστημα στο οποίο βασίστηκε και στήθηκε το σύστημα μας ήταν το Linux και πιο συγκεκριμένα η διανομή Ubuntu Server. Επίσης χρησιμοποιήθηκαν τα πρωτόκολλα HTTP, SSH για την επικοινωνία με την διεπαφή του συστήματος και για την επικοινωνία με τις εικόνες.

ABSTRACT

The task of this thesis is the development and creation of a cloud computing system, whose purpose is to share Operating System (OS) images to students of the department of Informatics at the Alexandrian Technological Educational Institute of Thessaloniki. The development of such an environment can have many applications which would ease the educational experience not only for students and personnel but also for the institute as a whole.

More specifically students can benefit from this system, by being enabled to have a personalized operating environment, at any point during the day, be it in the premises of the Institute, or through the use of a VPN. Institute personnel, can create separate OS images with the programs and services they desire, significantly improving the subject matter of each class. In addition the capacity of the computer labs would be increased, since it would be independent of the number of personal computers.

Finally, the CCS we created can be connected to, through the means of a simple browser. The user of this software, can create and distribute images through the network, or use other tools, creating a complete cloud computing experience.

We used the Openstack open source cloud software, and our OS was based on Linux; particularly on Ubuntu Server distribution. Finally, we used the HTTP and SSH protocols. HTTP was used in order to connect with the Web Interface of the system and the SSH protocol in order to have access not only to the command line of the bare OS but also to communicate with the images CLI.

Ευχαριστίες

Η πτυχιακή εργασία που εκπονήσαμε είναι αφιερωμένη στους δικούς μας ανθρώπους που βρίσκονται στο πλευρό μας και μας στηρίζουν με κάθε τρόπο, όχι μόνο κατά την διάρκεια εκπόνησης, αλλά σε κάθε πρόκληση και δυσκολία που αντιμετωπίζουμε.

Επίσης, ευχαριστήσουμε θερμά τον κ. Ηλιούδη για την ευκαιρία που μας έδωσε να ασχοληθούμε με ένα τόσο ενδιαφέρον και καινοτόμο θέμα και για την καθοδήγηση, τις προτάσεις και τις συμβουλές που μας παρείχε από την αρχή μέχρι την ολοκλήρωση της πτυχιακής.

Ακόμα, θέλουμε να ευχαριστήσουμε τον κ. Ψαρρά για την βοήθεια του σε δικτυακά ζητήματα και προβλήματα που προέκυπταν κατά την διεκπεραίωση της εργασίας μας.

Και τέλος, ευχαριστούμε τους φίλους μας, που ήταν πάντα εκεί, είτε με το να παρέχουν συμβουλές και προτάσεις πάνω στο αντικείμενο εκπόνησης, είτε να μας στηρίζουν ψυχολογικά για να ολοκληρώσουμε το έργο μας.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ 2
ΠΕΡΙΛΗΨΗ 3
ABSTRACT 5
ΠΕΡΙΕΧΟΜΕΝΑ 8
1 Εισαγωγή 15
1.1 Το πρόβλημα προς μελέτη 15
1.2 Σπουδαιότητα της εργασίας 15
1.3 Στόχος της εργασίας 16
1.4 Τα επιτεύγματα της πτυχιακής 16
1.5 Σύντομη περιγραφή κεφαλαίων 16
2 Το υπολογιστικό νέφος 18
Εισαγωγή 18
2.1 Ορισμός Υπολογιστικού Νέφους 18
2.2 Είδη υπηρεσιών Υπολογιστικού Νέφους 19
2.2.1 Διαθέσιμα μοντέλα υπηρεσιών Υπολογιστικού Νέφους 19
2.2.1.1 Infrastructure as a Service 19
2.2.1.2 Platform as a Service 21
2.2.1.3 Software as a Service 23
2.2.2 Διαθέσιμα μοντέλα υλοποίησης Υπολογιστικού Νέφους 24
2.2.2.1 Ιδιωτικό Νέφος (Private Cloud) 24
2.2.2.2 Δημόσιο Νέφος (Public Cloud) 25
2.2.2.3 Υβριδικό Νέφος (Hybrid Cloud) 26
2.3 Χαρακτηριστικά του Υπολογιστικού Νέφους 27
2.4 Ανησυχίες και παραπληροφόρηση που διέπουν το υπολογιστικό Νέφος 28
2.4.1 Όλα δουλεύουν καλύτερα στο Νέφος 28
2.4.2 Τα δεδομένα δεν είναι ασφαλή στη Νέφος 29
2.4.3 Είναι πάντα φτηνότερα να τρέχεις στο Νέφος 29
2.4.4 Όλα μπορούν να αυτοματοποιηθούν στο νέφος, έτσι δεν υπάρχει ανάγκη για υποστήριξη 29
2.4.5 Το Νέφος είναι επιβλαβές για το περιβάλλον 30
2.4.6 Το νέφος στερεί δουλειές 30
2.4.7 Η μετακίνηση στο Νέφος είναι πιο επιζήμια απ' ότι η αξία της 30

2.4.8 Τα «Big Data» δεν είναι μείζονος σημασίας. 31
2.4.9 Η τεχνολογία Νέφους είναι ακόμα στα σπάργανα. 31
2.5 Η Ασφάλεια στο Υπολογιστικό Νέφος 31
2.5.1 Στόχοι ασφαλείας του υπολογιστικού νέφους....32
2.5.2 Υπηρεσίες Ασφάλειας στο Νέφος 33
2.5.2.1 Αυθεντικοποίηση 33
2.5.2.2 Εξουσιοδότηση....33
2.5.2.3 Καταγραφή (Auditing)....34
2.5.2.4 Ευθύνη (Accountability)....35
Επίλογος 35
3 Περιγραφή των Τεχνολογικών Λύσεων 36
Εισαγωγή....36
3.1 Χαρακτηριστικά των διαθέσιμων επιλογών....37
3.1.1 CloudStack 37
3.1.1.1 Η ασφάλεια του CloudStack 38
3.1.2 Eucalyptus....42
3.1.2.1 Η ασφάλεια στο Eucalyptus....44
3.1.3 vCloud Director....46
3.1.3.1 Η ασφάλεια στο vCloud Director....47
3.1.4 Openstack 53
3.1.4.1 Η ασφάλεια στο Openstack 55
3.2 Η Σύγκριση των Επιλογών της 71
Επίλογος 80
4 Περιγραφή του Lab as a Service....82
Εισαγωγή....82
4.1 Περιγραφή του μοντέλου Lab-as-a-Service (LaaS)....82
4.2 Υφιστάμενες Lab-as-a-Service λύσεις 83
4.3 Virtual Computing Lab....84
Επίλογος 89
5 Περιγραφή του τεχνολογικού περιβάλλοντος Openstack....90
Εισαγωγή....90
5.1 Η αρχή και η εξέλιξη του Openstack....90
5.2 Τα Openstack Projects....94
5.3 Ανάπτυξη ενός Περιβάλλοντος Νέφους με το Openstack 95
5.4 Οι Βασικές Υπηρεσίες του Openstack 98
5.4.1 Keystone 99
5.4.2 Glance....110

5.4.3 Nova 119
5.4.4 Neutron 129
5.4.5 Cinder 140
5.4.6 Horizon (Dashboard) 149
Επίλογος 163
6 Το Περιβάλλον υλοποίησης 164
Εισαγωγή 164
6.1 Εξοπλισμός του μοντέλου Lab as a Service 164
6.1.1 Hardware 164
6.1.2 Software 165
6.2 Οι Clients 166
6.3 Χρήση του συστήματος ως απλός χρήστης (Web-Interface) 166
6.3.1 Είσοδος στο σύστημα 166
6.3.2 Ανάλυση δυνατοτήτων διεπαφής χρήστη 168
6.3.3 Δημιουργία Instances και αποθηκευτικού χώρου 173
6.3.3.1 Δημιουργία ενός νέου instance 173
6.3.3.2 Δημιουργία ενός νέου αποθηκευτικού χώρου 175
Επίλογος 177
7 Συμπεράσματα και μελλοντικές επεκτάσεις 178
Εισαγωγή 178
7.1 Συμπεράσματα 178
7.1.1 Συμπεράσματα ως προς τις δυνατότητες του συστήματος 178
7.1.2 Συμπεράσματα ως την λειτουργικότητα του συστήματος 179
7.1.3 Προστιθέμενη αξία 181
7.2 Μελλοντικές Επεκτάσεις 181
7.2.1 Επεκτάσεις ως προς την υπηρεσία 181
7.2.2 Επεκτάσεις ως προς την αλληλεπίδραση με άλλες υπηρεσίες του τμήματος 182
8 Ενδεικτική Βιβλιογραφία και Αναφορές 184
Ενδεικτική Βιβλιογραφία 184
Αναφορές 185

Ευρετήριο Εικόνων

2.1 Infrastructure-as-a-Service (IaaS) 20

2.2 “Platform-as-a-Service (PaaS)” 22

2.3 Ιδιωτικό Νέφος 25

2.4 Το Δημόσιο Νέφος 26

2.5 Το Υβριδικό Νέφος 27

3.1 Τα Λογισμικά υπολογιστικού νέφους που θα μελετήσουμε 36

3.2 Αυθεντικοποίηση με SAML 39

3.3 Διάγραμμα ροής του SAML 39

3.4 Ένας κανόνας εσωτερικής http πρόσβασης 41

3.5 Συμβατότητα Eucalyptus me Amazon API 43

3.6 Επισκόπηση της Αρχιτεκτονικής του Eucalyptus 44

3.7 Αρχιτεκτονική του VMWare vCloud Director 48

3.8 Η εμπιστοσύνη ανάλογα με το είδος του νέφους 56

3.9 Γεφύρωση Domains 58

3.10 API Endpoint Bridging 58

3.11 Τύποι Επιθέσεων 59

3.12 Κατηγορίες Επιθέσεων και πιθανότητα Επίθεσης και Πολυπλοκότητας 60

3.13 SSL/TLS proxy 63

3.14 Παράδειγμα Υλοποίησης Swift 65

3.15 Αρχιτεκτονική Αποθήκευσης Αντικειμένων με κόμβο Διαχείρισης – Object Storage Architecture Management (OSAM) 66

3.16 Αρχιτεκτονική και Ροή του Openstack Networking 68

3.17 CloudStack Management Console 76

3.18 vCloud Director Management Console 77

3.19 Openstack Management Console 77

4.1 Οι Υπηρεσίες του νέφους VCL 85

4.2 Η Φυσική Αρχιτεκτονική του VLC 86
4.3 Δημιουργία εικόνας μέσω web interface 87
4.4 Πρόσβαση μέσω SSH σε περιβάλλον Red Hat Linux 87
5.1 Το Openstack Foundation 91
5.2 Ο κύκλος έκδοσης Openstack / Openstack Release Cycle 93
5.3 Ημερομηνίες Εκδόσεων Openstack 94
5.4 Η δομή του Openstack 94
5.5 Πρώτος και Δεύτερος Τύπος Hypervisor 97
5.6 Η ροή Ταυτοποίησης του Openstack 100
5.7 Διαχείριση Glance Εικόνων εντός ενός Tenant 111
5.8 Regions, Aggregates και Availability Zones 121
5.9 Η Δομή της Υπηρεσίας Neutron 130
5.10 Cinder και φορητότητα των αποθηκευτικών χώρων 140
5.11 Η είσοδος στο Dashboard του περιβάλλοντος μας 150
5.12 Η Κεντρική Προεπισκόπηση του Horizon 151
5.13 Το Περιβάλλον Nova και οι Hypervisors 151
5.14 Τα Nova Instances που τρέχουν εντός του περιβάλλοντος 152
5.15 Η υπηρεσία Cinder 153
5.16 Flavors για τα προς δημιουργία Instances 154
5.17 Η Υπηρεσία Glance και η διαθέσιμες εικόνες 155
5.18 Οι προεπιλογές του συστήματος μας 155
5.19 Τα δίκτυα του Περιβάλλοντος 156
5.20 Οι δρομολογητές του Περιβάλλοντος 156
5.21 Οι ορισμοί των μεταδεδομένων 157
5.22 Πληροφορίες συστήματος και Endpoints 158
5.23 Πληροφορίες συστήματος και υπηρεσίες Nova 158
5.24 Πληροφορίες συστήματος και υπηρεσίες Cinder 159
5.25 Πληροφορίες συστήματος και υπηρεσίες Neutron 159

5.26 Τα διαθέσιμα Projects 160

5.27 Οι χρήστες 161

6.1 Η διεπαφή εισόδου του χρήστη 167

6.2 Η Κεντρική Προεπισκόπηση για τον χρήστη.....168

6.3 Το μενού Instances του χρήστη 169

6.4 Το μενού Volumes του χρήστη.....170

6.5 Το μενού Images του χρήστη.....170

6.6 Το μενού Access & Security του χρήστη 171

6.7 Το μενού με την τοπολογία του δικτύου 172

6.8 Το μενού με τα δίκτυα του περιβάλλοντος.....172

6.9 Το μενού με δρομολογητές του περιβάλλοντος 172

6. Επιλογές κατά την δημιουργία Image.....174

6.11 Επιλογές κατά την δημιουργία Volume.....176

7.1 Υλοποίηση Openstack με τρεις κόμβους.....180

Ευρετήριο πινάκων

Πίνακας 3.1 Υπηρεσίες αποθήκευσης……..…….....66

Πίνακας 3.2 Αλγόριθμοι κρυπτογράφησης και η χρηστικότητά τους……...69

Πίνακας 3.3 Σύγκριση Χαρακτηρίστικών……...71

1 Εισαγωγή

1.1 Το πρόβλημα προς μελέτη

Η συγκεκριμένη πτυχιακή εργασία πραγματεύεται ένα συγκεκριμένο πρόβλημα. Την εξομοίωση ενός εργαστηριακού περιβάλλοντος με την χρήση της τεχνολογίας του υπολογιστικού νέφους[1]. Πρακτικά, την μεταφορά των λειτουργικών συστημάτων που φιλοξενούνται στους τωρινούς ηλεκτρονικούς υπολογιστές των εργαστηριακών τμημάτων στην υποδομή του νέφους. Επίσης, μελετάται όλη η διαδικασία σχεδίασης και ανάπτυξης ενός τέτοιου συστήματος καθώς και η λειτουργικότητά του.

1.2 Σπουδαιότητα της εργασίας

Με την υλοποίηση του συγκεκριμένου εγχειρήματος, δεν θα είναι απαραίτητη η ύπαρξη του εργαστηριακού εξοπλισμού των τμημάτων. Πλέον δεν θα υπάρχουν ηλεκτρονικοί υπολογιστές που θα φιλοξενούν τα διάφορα λειτουργικά συστήματα που χρησιμοποιούνται στα πλαίσια των μαθημάτων. Θα υπάρχει απλά ο χώρος όπου ο φοιτητής θα έρχεται με τον προσωπικό του ηλεκτρονικό υπολογιστή και αφού συνδεθεί στο σύστημα μας θα μπορεί να δουλεύει κανονικά όπως συμβαίνει με την τωρινή υποδομή του τμήματος. Παράλληλα θα μπορεί να σώζει την πρόοδο της εργασίας του, στον προσωπικό αποθηκευτικό του χώρο, και να την συνεχίζει είτε στο επόμενο εργαστηριακό του μάθημα είτε από τον χώρο της αρεσκείας του, απλά συνδεόμενος στο δίκτυο του ΤΕΙ με την χρήση του VPN[2].

Επίσης, το σύστημα που δημιουργήσαμε βασίζεται σε λογισμικά ανοιχτού κώδικα. Άμεση συνέπεια αυτού είναι ότι μπορεί να τροποποιηθεί, να επεκταθεί και να βελτιωθεί στο μέλλον.

Η έρευνα και η ενασχόληση με τον συγκεκριμένο τεχνολογικό τομέα, δίνει την δυνατότητα εκμάθησης και εξοικείωσης τόσο με το λογισμικό του Openstack[3] που αποτελεί ένα από τα καλύτερα εργαλεία στον χώρο του αλλά και με την διαχείριση ενός Linux[4] Server βασισμένο στην Ubuntu[5] διανομή.

1.3 Στόχος της εργασίας

Στόχος της συγκεκριμένης πτυχιακής εργασίας είναι η μελέτη και η ανάπτυξη ενός συστήματος που θα βασίζεται στο υπολογιστικό νέφος. Σκοπός της δημιουργίας του συστήματος αυτού θα είναι να φιλοξενεί εικονικά λειτουργικά συστήματα[6] και στην συνέχεια να τα διαμοιράζει προς τους φοιτητές και το εκπαιδευτικό προσωπικό για την διεκπεραίωση του εκάστοτε εργαστηριακού μαθήματος.

Ειδικότερα, ο ενδιαφερόμενος θα εισέρχεται στο σύστημα με προσωπικά στοιχεία ταυτοποίησης και στην συνέχεια θα μπορεί να δουλεύει πάνω σε ένα εικονικό λειτουργικό σύστημα ενός εικονικού μηχανήματος έχοντας παράλληλα και τον προσωπικό του αποθηκευτικό χώρο που θα τα συνοδεύει.

1.4 Τα επιτεύγματα της πτυχιακής

Το σημαντικότερο επίτευγμα της πτυχιακής ήταν ότι λειτούργησε ένα σύστημα με την χρήση λογισμικού υπολογιστικού νέφους. Έγινε εφικτό να δημιουργηθούν διαφορετικοί χρήστες που ο καθένας θα έχει διαφορετικό ρόλο και δικαιώματα κατά την χρησιμοποίηση του συστήματος.

1.5 Σύντομη περιγραφή κεφαλαίων

Στο αμέσως επόμενο κεφάλαιο γίνεται μια γενική περιγραφή του υπολογιστικού νέφους. Στην συνέχεια, αναφέρονται τα είδη των υπηρεσιών του υπολογιστικού νέφους και τα μοντέλα υλοποίησης του. Επίσης, γίνεται αναφορά στα χαρακτηριστικά του υπολογιστικού νέφους καθώς και στην ασφάλεια του και τα χαρακτηριστικά αυτής.

Στο τρίτο κεφάλαιο γίνεται μια περιγραφή των υφιστάμενων τεχνολογικών λύσεων. Αναλύονται και αξιολογούνται τέσσερις ήδη υπάρχουσες λύσεις στον χώρο του υπολογιστικού νέφους. Εκτός από την ανάλυση των χαρακτηριστικών γίνεται εκτενής αναφορά και ανάλυση της ασφάλειας που χαρακτηρίζει την κάθε λύση. Τέλος, συγκρίνονται οι τέσσερις αυτές επιλογές και καταλήγουμε στην επιλογή του λογισμικού που επιλέξαμε για να χρησιμοποιήσουμε.

Στο τέταρτο κεφάλαιο γίνεται περιγραφή του Lab as a Service. Περιγράφεται η φιλοσοφία γύρω από το εγχείρημα αυτό πώς το σύστημα θα

χρησιμοποιείται. Στην συνέχεια γίνεται αναφορά στις ήδη υπάρχουσες λύσεις και αναλύεται η πιο διαδεδομένη και λειτουργική από αυτές.

Το πέμπτο κεφάλαιο πραγματεύεται την περιγραφή του τεχνολογικού περιβάλλοντος με το οποίο ασχοληθήκαμε. Γίνεται εκτενής περιγραφή του λογισμικού, αναφέρεται η εξέλιξη του και περιγράφονται τα χαρακτηριστικά του. Επίσης γίνεται ανάλυση των υπηρεσιών του λογισμικού και παράλληλα η ανάπτυξη της κάθε υπηρεσίας στα πλαίσια του συστήματος. Η αλληλεπίδραση των υπηρεσιών αυτών έχει σαν αποτέλεσμα την λειτουργικότητα του συστήματος. Τέλος, στα πλαίσια της ανάλυσης της τελευταίας υπηρεσίας περιγράφεται και το περιβάλλον εργασίας του διαχειριστή του συστήματος.

Στο έκτο κεφάλαιο γίνεται περιγραφή του περιβάλλοντος υλοποίησης. Αναφέρεται ο υλικός εξοπλισμός που χρησιμοποιήθηκε για την πτυχιακή εργασία. Επίσης αναφέρονται το λογισμικό καθώς και οι clients για την χρήση του συστήματος. Τέλος, περιγράφεται η εμπειρία και η διαδικασία χρήσης του συστήματος από έναν απλό χρήστη και η δυνατότητες που αυτός έχει.

2 Το υπολογιστικό νέφος

Εισαγωγή

Στο κεφάλαιο αυτό θα μελετήσουμε την τεχνολογία που διέπει το υπολογιστικό νέφος. Θα δούμε πως διαχωρίζονται τα υπολογιστικά νέφη και θα μελετήσουμε τα διαφορετικά ήδη του διαχωρισμού αυτού. Στην συνέχεια, θα μελετήσουμε την ασφάλεια του νέφους μαζί με τους στόχους που θα πρέπει να επιτευχθούν έτσι ώστε να θεωρείται ένα νέφος πλήρως ασφαλές. Στο πλαίσιο της ασφάλειας θα αναφερθούν και οι υπηρεσίες που προσφέρονται στο νέφος.

Στο τέλος αυτού του κεφαλαίου φιλοδοξούμε να υπάρχει μια ξεκάθαρη εικόνα για το υπολογιστικό νέφος αν και θα πρέπει να διευκρινιστεί ίσως από την αρχή πως οι απόψεις διαφορετικών ανθρώπων όταν ερωτηθούν «Τι είναι υπολογιστικό νέφος» θα είναι πάντα ασύμφωνες δεδομένου του είδους νέφους με το οποίο έχουν έρθει οι ίδιοι σε επαφή καθώς και με την κατηγορία των παρεχόμενων υπηρεσιών αυτού.

2.1 Ορισμός Υπολογιστικού Νέφους

Ίσως ο πιο απλός και λειτουργικός - πλην όμως γενικευμένος - ορισμός του νέφους είναι «η δυνατότητα πρόσβασης μέσω φυλλομετρητή σε αρχεία, δεδομένα και υπηρεσίες τα οποία φιλοξενούνται από έναν πάροχο».

Πολύ συχνά ο όρος νέφος χρησιμοποιείται ως συνώνυμο, λανθασμένα όπως θα δούμε παρακάτω, με όρους όπως utility computing, software-as-a-service (SaaS), και grid computing. Από όλα αυτά το utility computing και το SaaS είναι απλά δύο από τις πολλές μορφές υπηρεσιών που μπορεί να προσφέρει ένα νέφος. Όσο για το grid computing είναι απλά ένας τύπος υποκείμενων τεχνολογιών πάνω στις οποίες μπορεί να υλοποιηθεί ένα νέφος.

Ο όρος «νέφος» χρησιμοποιείται επίσης πολλές φορές συνώνυμα με το «data center». Με την πρόοδο όμως της τεχνολογίας ένα νέφος είναι πολλά περισσότερα από ένα data center. Αξιοποιώντας την πρόοδο στην ραχοκοκαλιά του Διαδικτύου, την ευρεία πρόσβαση σε αυτό, τους πανίσχυρους εξυπηρετητές και τον άπλετο χώρο αποθήκευσης στα data centers καθώς και την επεκτασιμότητα του λογισμικού αυτών καταφέρνει να συνδυάσει ένα μοναδικό σύνολο δυνατοτήτων για προσφορά στους χρήστες του.

Θα πρέπει να θυμόμαστε πως το νέφος δεν αποτελεί μια τεχνολογία από μόνο του. Αποτελεί μια προσέγγιση στην δημιουργία υπηρεσιών πληροφορικής, οι οποίες έχουν την δυνατότητα να αξιοποιήσουν στο μέγιστο το hardware αλλά και τις τεχνολογίες εικονικοποίησης (virtualization)[7], είτε συνδυάζοντας πολλούς εξυπηρετητές (servers), είτε διαιρώντας ενιαίους εξυπηρετητές σε ξεχωριστά εικονικά συστήματα, τα οποία έχουν την δυνατότητα εκκίνησης και τερματισμού κατά το δοκούν.

2.2 Είδη υπηρεσιών Υπολογιστικού Νέφους

Το Υπολογιστικό Νέφος όπως αναφέραμε και προηγουμένως μπορεί να κατανεμηθεί σε **δυο υποκατηγορίες**: ως προς το **είδος της υπηρεσίας** που προσφέρεται και ως προς το **μοντέλο υλοποίησης (deployment model)**.

Ξεκινώντας από τα είδη των υπηρεσιών, τα διαθέσιμα μοντέλα του Υπολογιστικού Νέφους είναι τα Software-as-a-Service, Platform-as-a-Service και Infrastructure-as-a-Service. Το κάθε ένα από αυτά, εξυπηρετεί διαφορετικές ανάγκες και προσφέρει διαφορετικές υπηρεσίες.

2.2.1 Διαθέσιμα μοντέλα υπηρεσιών Υπολογιστικού Νέφους

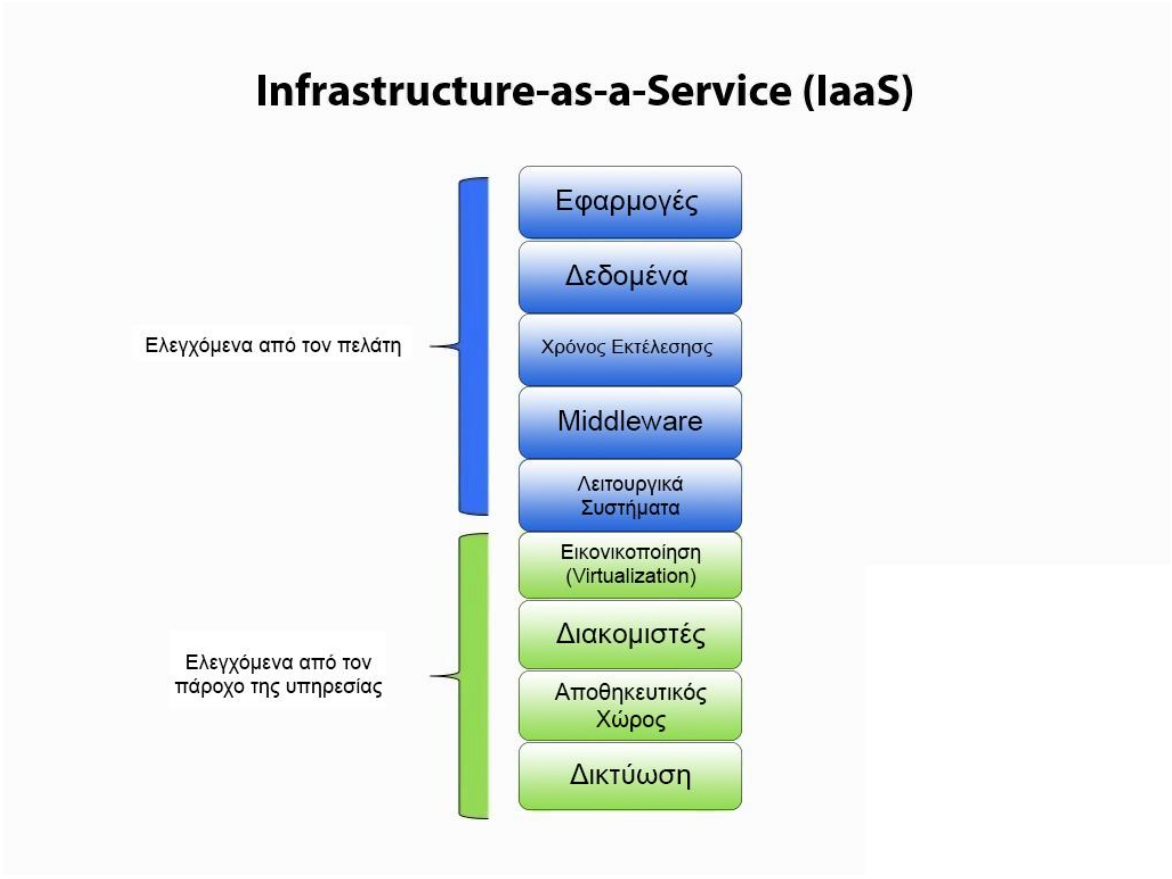
2.2.1.1 Infrastructure as a Service

Infrastructure-as-a-Service (IaaS)[8] είναι η υπηρεσία διάθεσης υλικοτεχνικής υποδομής (hardware) όπως διακομιστές (servers), αποθηκευτικά μέσα (storage), υποδομές δικτύου (network) και σχετικού με αυτά λογισμικό όπως λειτουργικά συστήματα (operating systems), τεχνολογίες εξομοίωσης (virtualization technology) και συστήματα διαχείρισης αρχείων (file systems). Πρόκειται για μια εξέλιξη του μοντέλου της παραδοσιακής φιλοξενίας αρχείων, η οποία όμως στην νέα της μορφή δεν απαιτεί μακροχρόνιες συμβάσεις ενώ επιπλέον επιτρέπει στους τελικούς χρήστες να δεσμεύουν και να χρησιμοποιούν δυναμικά τις υποδομές ανάλογα με την ζήτηση. Στο μοντέλο αυτό ο πάροχος IaaS εκτελεί πολύ λίγες εργασίες υποστήριξης και οι τελικοί χρήστες πρέπει να διαμοιράσουν και να διαχειριστούν τις υπηρεσίες με τον ίδιο ακριβώς τρόπο, όπως θα έκαναν αν οι υποδομές ήταν στο δικό τους datacenter.

Το μοντέλο Infrastructure as a Service είναι μια μορφή φιλοξενίας (hosting). Περιλαμβάνει πρόσβαση στο δίκτυο καθώς και υπηρεσίες δρομολόγησης και

αποθήκευσης δεδομένων. Ο πάροχος IaaS σε γενικές γραμμές συντηρεί το υλικό και διαχειρίζεται τις υπηρεσίες (services) που είναι αναγκαίες για να «τρέξουν» τα λογισμικά. Η δυναμική αυξομείωση του εύρους ζώνης (bandwidth), της μνήμης, και του αποθηκευτικού χώρου είναι ένα από τα κύρια χαρακτηριστικά του μοντέλου. Η ταχύτητα της δυναμικής προσαρμογής και η τιμολόγηση της αποτελεί πεδίο ισχυρού ανταγωνισμού μεταξύ των παρόχων υπηρεσιών IaaS.

Δεκάδες εκατοντάδες εφαρμογές που υποστηρίζουν παράλληλη χρήση σε διάφορες συσκευές (smart phones, tablets, desktop applications) εμπίπτουν στην κατηγορία αυτή. Τα δεδομένα τους είναι αποθηκευμένα σε υποδομές του νέφους και διαθέσιμα προς χρήση στην τελευταία τους μορφή από όποια συσκευή και αν ζητηθούν.



2.1 Infrastructure-as-a-Service (IaaS)

2.2.1.2 Platform as a Service

Το IaaS το ακολουθεί το Platform-as-a-Service[9]. Με το Platform-as-a-Service, ο Πάροχος παρέχει περισσότερα από υποδομή. Παρέχει αυτό που θα αποκαλούσαμε μια ενοποιημένη λύση – ένα πλήρες σύνολο λογισμικού που δίνει την δυνατότητα σε έναν προγραμματιστή να το χρησιμοποιήσει για να φτιάξει το οτιδήποτε – τόσο για ανάπτυξη εφαρμογών λογισμικού αλλά και περιβάλλοντα χρόνου εκτέλεσης.

Το PaaS μπορεί να θεωρηθεί και ως εξέλιξη του web hosting. Τα τελευταία χρόνια, οι εταιρείες web hosting παρέχουν ένα αρκετά πλήρες «πακέτο λογισμικού» για την ανάπτυξη Web Sites. Το PaaS πάει αυτή την ιδέα ένα βήμα μετά παρέχοντας διαχείριση του κύκλου ζωής – την δυνατότητα για την διαχείριση όλων των σταδίων ανάπτυξης λογισμικού από τον προγραμματισμό και σχεδιασμό, στη δημιουργία και την ανάπτυξη, ως τον έλεγχο και την συντήρηση.

Το κύριο πλεονέκτημα του PaaS είναι ότι έχει δυνατότητα ανάπτυξης και εκτέλεσης λογισμικού βασισμένη αποκλειστικά στο Νέφος – ως εκ τούτου, δεν απαιτούνται προσπάθειες διαχείρισης και συντήρησης για την υποδομή. Κάθε πλευρά της ανάπτυξης λογισμικού, από το στάδιο σχεδιασμού και μετά (συμπεριλαμβανομένης της διαχείρισης πηγαίου κώδικα, του ελέγχου και της εκτέλεσης) υπάρχει στο Νέφος.

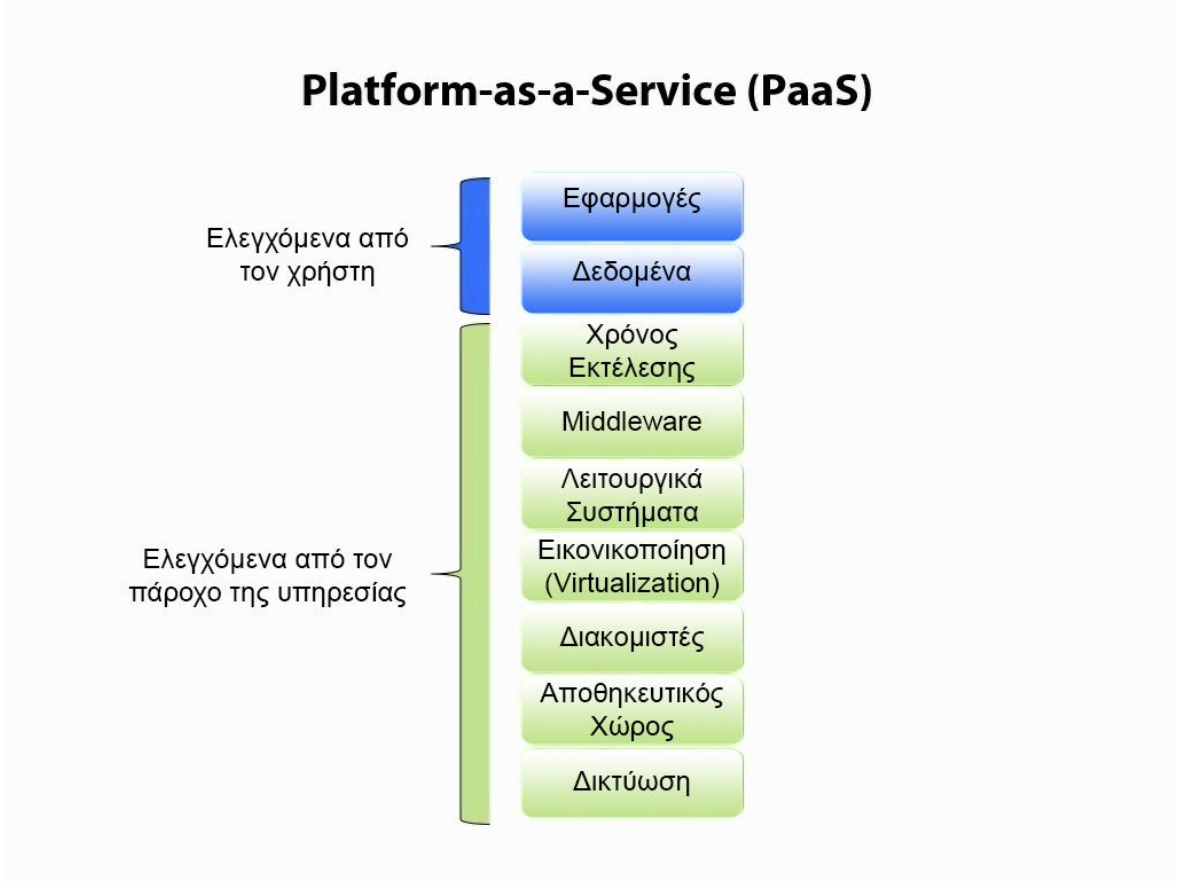
Το PaaS είναι πολύ-γειτονικό κάτι που σημαίνει πως φυσικά υποστηρίζει όλο το πακέτο των θεμελιωδών Web services και συνήθως παρέχεται με δυναμική κλιμάκωση. Κατ’ αντιστοιχία με προηγουμένως, δυναμική κλιμάκωση σημαίνει ότι το λογισμικό μπορεί να αυξηθεί ή να μειωθεί αυτομάτως. Το Platform-as-a-Service καλύπτει την ανάγκη για κλιμάκωση όπως επίσης και την ανάγκη για διευθέτηση σοβαρών θεμάτων όπως προσβασιμότητα και ασφάλεια δεδομένων για τους χρήστες του.

Παρόλα τα πλεονεκτήματα αυτής της προσέγγισης σαφώς υπάρχουν και μειονεκτήματα. Το κύριο μειονέκτημα του Platform-as-a-Service είναι ότι μπορεί να μας περιορίσει στη χρήση ενός συγκεκριμένου περιβάλλοντος ανάπτυξης και ενός συγκεκριμένου συνόλου προγραμματιστικών εργαλείων. Κατά συνέπεια μπορεί να βρεθούμε «παντρεμένοι» με την πλατφόρμα που παρέχει ο Πάροχος και ανίκανοι να μεταφέρουμε τις εφαρμογές μας οπουδήποτε αλλού χωρίς να χρειαστεί να

ξαναγράψουμε εξ αρχής μεγάλο μέρος του κώδικα. Αν για κάποιο λόγο λοιπόν, μείνουμε δυσαρεστημένοι από τον Πάροχο PaaS που επιλέξαμε, υπάρχει μεγάλο ενδεχόμενο αντιμετώπισης υψηλών εξόδων αφού θα πρέπει να ξαναγράψουμε πολύ ή ολόκληρο τον κώδικά μας για να ανταποκριθούμε στις απαιτήσεις του νέου PaaS παρόχου που θα έχουμε επιλέξει.

Ο φόβος για αυτό τον εγκλωβισμό που αναφέραμε παραπάνω οδήγησε στην ανάπτυξη μιας νέας κατηγορίας PaaS: το Open Platform-as-a-Service. Αυτό θα προσφέρει την ίδια προσέγγιση όπως το PaaS, μόνο που δεν θα υπάρχει περιορισμός επιλογής στο λογισμικό ανάπτυξης.

Μερικά παραδείγματα PaaS είναι το Google App Engine, το AppJet, το Etelos, το Qrimp.com και το Force.com το οποίο αποτελεί το επίσημο περιβάλλον ανάπτυξης για το Salesforce.com.



2.2 “Platform-as-a-Service (PaaS)”

2.2.1.3 Software as a Service

Τέλος στο Software-as-a-Service[10], δίνεται πρόσβαση στους χρήστες σε εφαρμογές λογισμικού και βάσεων δεδομένων. Οι πάροχοι νέφους διαχειρίζονται την υποδομή και την πλατφόρμα που τρέχει αυτές τις εφαρμογές. Στο SaaS αναφερόμαστε και κάποιες φορές ως «λογισμικό κατ' απαίτηση» και χρεώνεται συνήθως με βάση το ποσοστό χρήσης ή με κάποιο συνδρομητικό ποσό.

Στο μοντέλο SaaS οι πάροχοι νέφους εγκαθιστούν και χειρίζονται τις εφαρμογές λογισμικού στο νέφος και οι χρήστες έχουν πρόσβαση στο λογισμικό αυτό μέσω των clients νέφους. Οι χρήστες του νέφους δεν επεμβαίνουν στην υποδομή και την πλατφόρμα στην οποία τρέχει το λογισμικό. Αυτό εξουδετερώνει την ανάγκη να εγκατασταθεί και να τρέξει κάποια εφαρμογή στον cloud-client του χρήστη, κάτι το οποίο με τη σειρά του απλοποιεί την συντήρηση και την υποστήριξη. Οι εφαρμογές νέφους είναι διαφορετικές σε ότι αφορά την επεκτασιμότητα τους – κάτι το οποίο μπορεί να επιτευχθεί με κλωνοποίηση διεργασιών σε πολλαπλά εικονικά μηχανήματα κατά την διάρκεια εκτέλεσης για να ικανοποιηθεί ο εργασιακός φόρτος. Οι ισορροπιστές φόρτου (load balancers) διαμοιράζουν τις διεργασίες σε μια ομάδα εικονικών μηχανημάτων. Αυτή η διαδικασία είναι διαφανής στον χρήστη, ο οποίος βλέπει ένα και μοναδικό σημείο πρόσβασης.

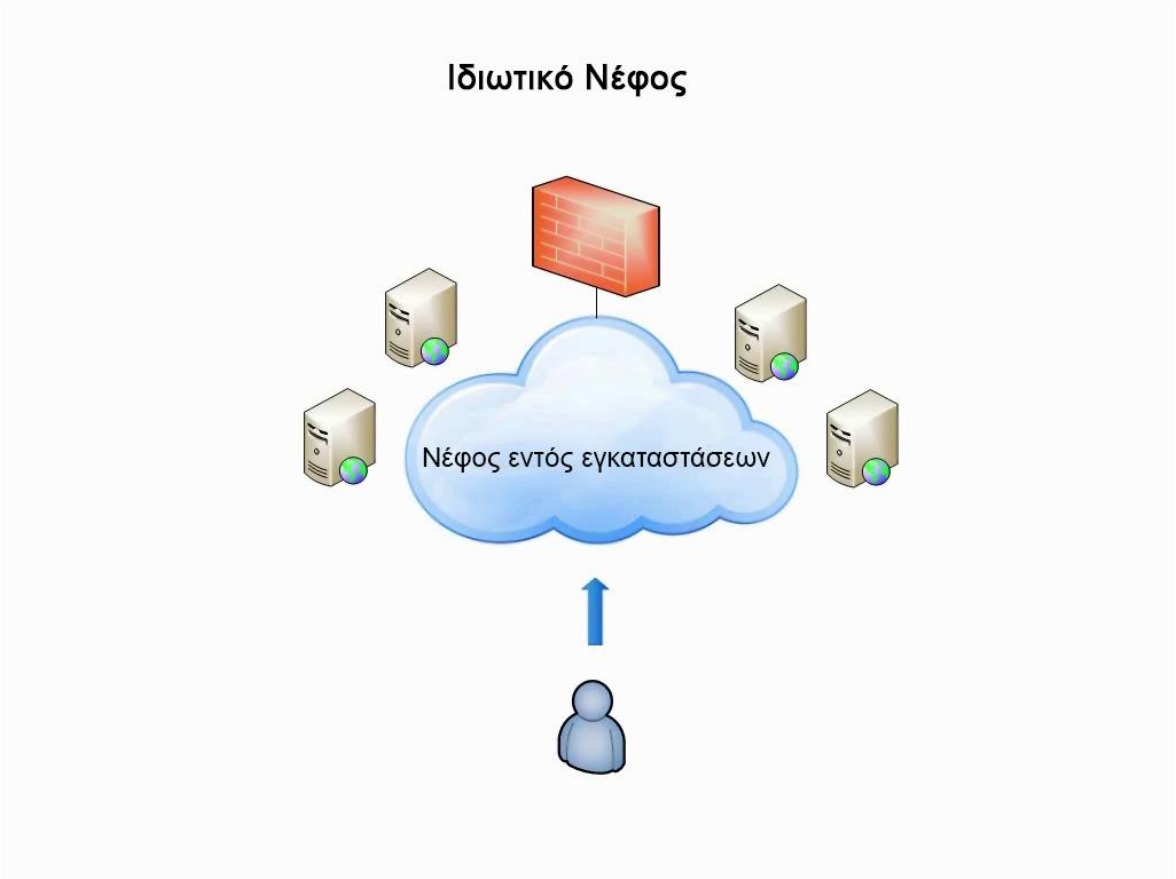
Οι υποστηρικτές ισχυρίζονται πως το SaaS επιτρέπει στις επιχειρήσεις να μειώσουν τις λειτουργικές δαπάνες πληροφορικής με το να μεταθέτουν την συντήρηση και υποστήριξη του υλικού και του λογισμικού του νέφους στον εκάστοτε πάροχο. Αυτό συντελεί στην μείωση των δαπανών των εταιρειών πληροφορικής για συντήρηση εξοπλισμού και την επένδυση αυτών προς άλλους στόχους. Ένα βασικό μειονέκτημα του SaaS είναι ότι τα δεδομένα των χρηστών καταχωρούνται στον εξυπηρετητή του παρόχου νέφους. Κατά συνέπεια θα μπορούσε να υπάρχει μη εξουσιοδοτημένη πρόσβαση σε αυτά τα δεδομένα.

2.2.2 Διαθέσιμα μοντέλα υλοποίησης Υπολογιστικού Νέφους

Συνεχίζοντας με τα μοντέλα υλοποίησης, τα διαθέσιμα μοντέλα υλοποίησης Νέφους είναι το **Ιδιωτικό Νέφος (Private Cloud)**, το **Δημόσιο Νέφος (Public Cloud)** και το **Υβριδικό Νέφος (Hybrid Cloud)**.

2.2.2.1 Ιδιωτικό Νέφος (Private Cloud)

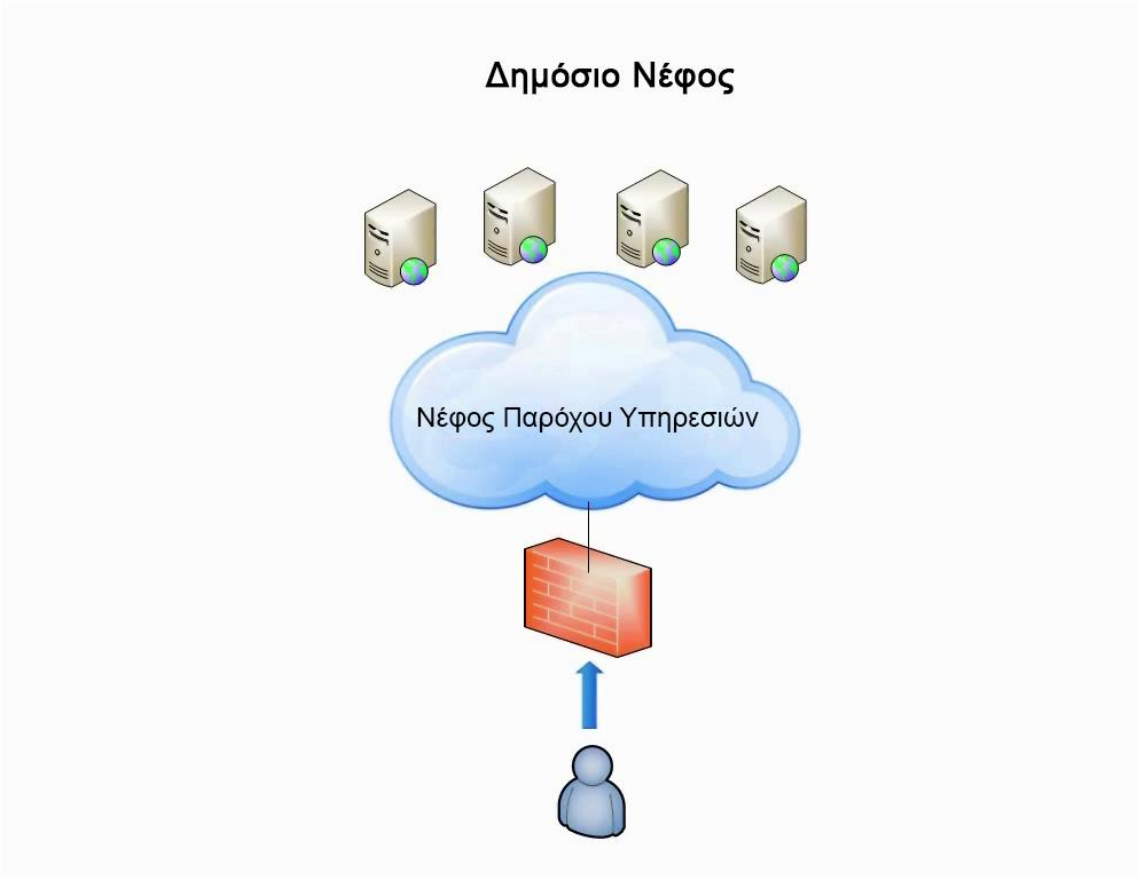
Το ιδιωτικό Νέφος[11] είναι μια δομή νέφους που τρέχει αποκλειστικά για έναν και μόνο οργανισμό, ο χειρισμός του γίνεται είτε εσωτερικά είτε από κάποιον τρίτο και φιλοξενείται εσωτερικά (στον χώρο της εταιρείας/οργανισμού) ή εξωτερικά. Η ανάληψη υλοποίησης ενός ιδιωτικού νέφους απαιτεί ένα ορισμένο επίπεδο και βαθμό ενασχόλησης για την εικονικοποίηση του επιχειρηματικού περιβάλλοντος, και απαιτεί ο οργανισμός να επανεξετάσει αποφάσεις για ήδη υπάρχουσα υποδομή. Αν υλοποιηθεί σωστά, μπορεί να βελτιώσει την αποδοτικότητα της εργασίας, αλλά κάθε βήμα σε αυτή την υλοποίηση ανεγείρει θέματα ασφαλείας που πρέπει να διευθετηθούν για να αποτραπούν σοβαρές ευπάθειες. Τα ιδιόκτητα και ιδιοσυντηρούμενα data centers είναι συνήθως πολύ δαπανηρά. Έχουν σημαντικό αντίκτυπο, αφού απαιτούν αρκετή έκταση χώρου, υπολογιστικού υλικού, και ελέγχου περιβάλλοντος. Όλα τα προηγούμενα πρέπει να ανανεώνονται περιοδικά, κάτι που συμβάλλει σε επιπλέον έξοδα. Έχουν προσελκύσει αρνητική κριτική επειδή «οι χρήστες εξακολουθούν να πρέπει να αγοράζουν και να συντηρούν».



2.3 Ιδιωτικό Νέφος

2.2.2.2 Δημόσιο Νέφος (Public Cloud)

Ένα νέφος καλείται «Δημόσιο Νέφος»[12] όταν οι υπηρεσίες του καθίστανται διαθέσιμες μέσω ενός δικτύου το οποίο είναι ανοικτό για δημόσια χρήση. Οι υπηρεσίες του δημόσιου νέφους μπορεί να είναι δωρεάν ή να προσφέρονται με μία συνθήκη πληρωμής σύμφωνα με τη χρήση τους. Τεχνικά μπορεί να υπάρχει λίγη έως καθόλου διαφορά μεταξύ ενός δημόσιου και ενός ιδιωτικού νέφους όσον αφορά στην αρχιτεκτονική, ωστόσο, η ενασχόληση γύρω από την ασφάλεια μπορεί να είναι σημαντικά διαφορετική για τις υπηρεσίες (εφαρμογές, αποθηκευτικός χώρος, άλλες υποδομές) που είναι διαθέσιμες από έναν πάροχο για δημόσιο κοινό και όταν η επικοινωνία διεξάγεται μέσω ενός μη-έμπιστου δικτύου. Γενικά, πάροχοι υπηρεσιών δημοσίου νέφους όπως η Amazon AWS, η Microsoft και η Google έχουν δική τους ιδιόκτητη υποδομή στα data center τους την οποία και διαχειρίζονται και η πρόσβαση σε αυτή την υποδομή γίνεται γενικά μέσω Internet.



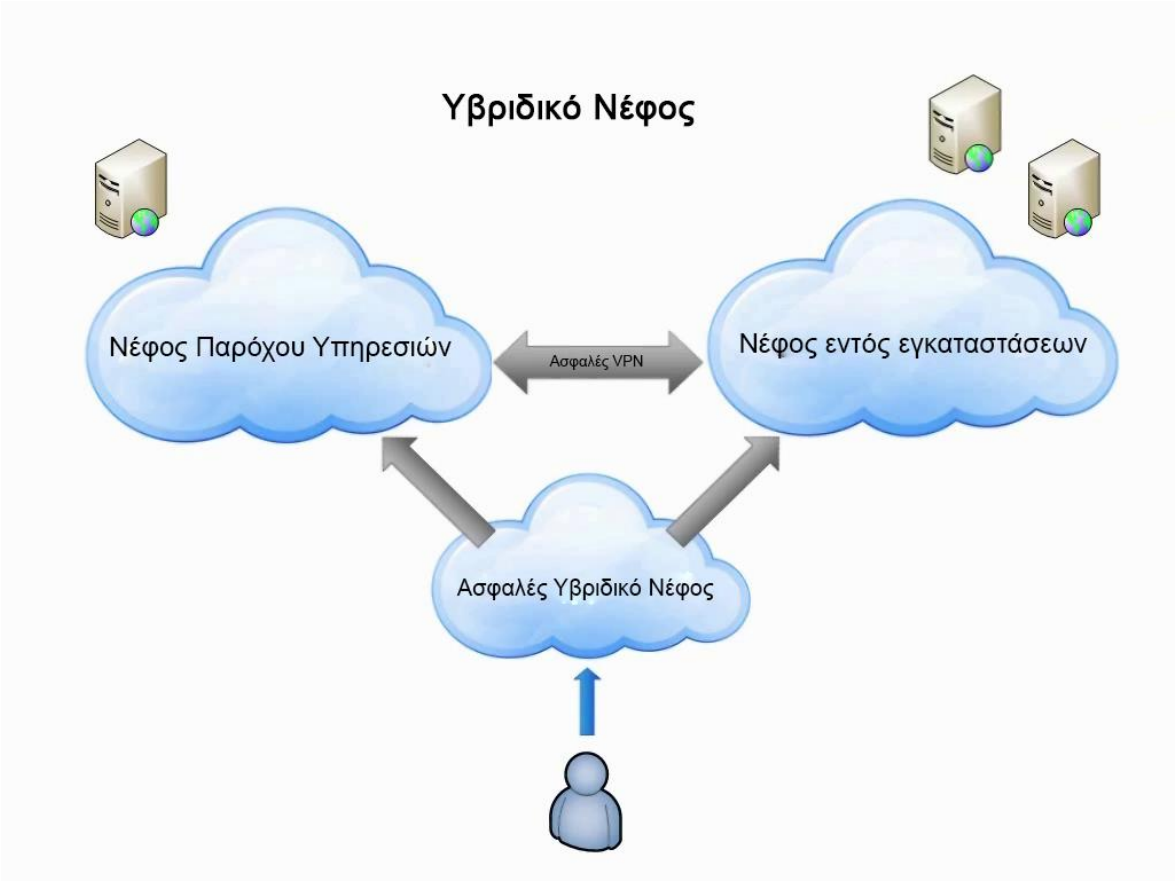
2.4 Το Δημόσιο Νέφος

2.2.2.3 Υβριδικό Νέφος (Hybrid Cloud)

Υβριδικό Νέφος[13] είναι μία σύνθεση δύο ή περισσότερων νεφών (ιδιωτικά ή δημόσια) τα οποία παραμένουν ξεχωριστές οντότητες αλλά είναι αλληλένδετα συνδεδεμένα μαζί, προσφέροντας τα πλεονεκτήματα πολλών μοντέλων υλοποίησης. Το Υβριδικό νέφος μπορεί επίσης να ερμηνευθεί ως η δυνατότητα να συνδέσεις (στην ίδια τοποθεσία), διαφόρων ειδών services με υποδομές νέφους.

Ένα παράδειγμα υβριδικού νέφους είναι ένας οργανισμός πληροφορικής που χρησιμοποιεί υποδομή δημόσιου νέφους για να ανταπεξέλθει σε προσωρινές ανάγκες χωρητικότητας που δεν μπορούν να καλυφθούν από ένα ιδιωτικό νέφος. Αυτή η δυνατότητα επιτρέπει στα υβριδικά νέφη να υιοθετήσουν το μοντέλο “cloud bursting” για κλιμάκωση μεταξύ νεφών. Το μοντέλο cloud bursting είναι ένα μοντέλο ανάπτυξης λογισμικού κατά το οποίο μια εφαρμογή τρέχει σε ένα ιδιωτικό νέφος ή data center και «εκτοξεύεται» (burst) σε ένα δημόσιο νέφος όταν η ζήτηση για υπολογιστική ισχύ αρχίσει να αυξάνεται. Ένα κύριο πλεονέκτημα του cloud

bursting και του υβριδικού μοντέλου νέφους είναι ότι ο οργανισμός πληρώνει μόνο την επιπλέον υπολογιστική υποδομή όταν αυτή χρειάζεται.



2.5 Το Υβριδικό Νέφος

2.3 Χαρακτηριστικά του Υπολογιστικού Νέφους

Τα χαρακτηριστικά του Υπολογιστικού Νέφους που προσφέρονται από τους παρόχους είναι:

- **Scalability:** Ανακατανομή δεδομένων καθώς εισάγεται νέο υλικό.
- **Virtualization:** Δυνατότητα εικονικών μηχανών.
- **Pay as you Go / Pay as you Grow:** Πληρωμή ανάλογα με τη ζήτηση πόρων και μέσων και την ανάπτυξη των αναγκών του πελάτη.
- **Multitenancy:** Δυνατότητα υποστήριξης διαφορετικών εφαρμογών.
- **Elasticity:** Δυνατότητα λήψης επιπλέων πόρων στις διάφορες εφαρμογές που υποστηρίζονται και πλήρη κάλυψη των αναγκών που προκύπτουν.
- **Load and Tenant Balancing:** Δυνατότητα μεταφοράς φορτίου ανάμεσα στους εξυπηρετητές για την αποφυγή υπερφόρτωσης.

- **Availability:** Συνέχιση λειτουργίας συστήματος ακόμα και σε περίπτωση υψηλών ποσοστών αποτυχίας εξυπηρετητών χωρίς την «πτώση» των υπηρεσιών που παρέχονται.
- **Security:** Μεγάλη ασφάλεια για την αδιάλειπτη λειτουργία των εφαρμογών χωρίς κανένα πρόβλημα.
- **Metering:** Δυνατότητα παρακολούθησης της χρήσης των πόρων που προσφέρονται καθώς και λήψη ειδοποιήσεων όταν κάποιος πόρος φτάσει σε οριακό σημείο και πρέπει να αυξηθεί.
- **Global:** Δυνατότητα χρήσης των υπηρεσιών νέφους από παντού.
- **Simple APIs:** Διευκόλυνση ανάπτυξης των εφαρμογών που χρησιμοποιούνται για υπηρεσίες νέφους.

2.4 Ανησυχίες και παραπληροφόρηση που διέπουν το υπολογιστικό Νέφος

Πολλοί φορείς αλλά και ιδιώτες και πελάτες σκέφτονται πριν αποφασίσουν να μεταβούν στην τεχνολογία του νέφους. Σε αυτό το σημείο θα παραθέσουμε τις συχνότερες ανησυχίες αλλά και τις περιπτώσεις παραπληροφόρησης[14] που αιωρούνται γύρω από το υπολογιστικό νέφος.

2.4.1 Όλα δουλεύουν καλύτερα στο Νέφος

Εδώ η αλήθεια βρίσκεται κάπου στη μέση. Το νέφος είναι μεν ιδανικό για τις περισσότερες επιχειρήσεις, δεν έχει σημασία αν είναι μεγάλες πολυεθνικές, οικονομικά ινστιτούτα, online επιχειρήσεις ή μια απλή επιχείρηση. Τα οφέλη της ταχύτητας προς την αγορά, απλοποιημένων διαδικασιών και ελαστικών κοστών υποδομής είναι δύσκολο να τα αγνοήσουμε. Αλλά το γεγονός ότι το Νέφος είναι καλό για σχεδόν κάθε επιχείρηση δεν σημαίνει αυτόματα ότι είναι και σωστό για κάθε εργασία που θα θέλαμε να κάνει η πληροφοριακή μας υποδομή. Το δημόσιο νέφος όπως είδαμε και παραπάνω, μπορεί να είναι σωστό για κάποιες δραστηριότητες, το ιδιωτικό νέφος για κάποιες άλλες και το αποκλειστικό hosting για παλιές εφαρμογές. Λαμβάνεται έτσι η συνολικά καλύτερη λύση με έναν συνδυασμό δύο ή τριών επιλογών από τις παραπάνω. Για παράδειγμα εάν τρέξουμε μια παλιά εφαρμογή που σχεδιάστηκε να τρέχει σε dedicated servers, τότε η μετακίνησή της στο νέφος μπορεί συχνά να αποδειχθεί δύσκολη.

2.4.2 Τα δεδομένα δεν είναι ασφαλή στη Νέφος

Η ασφάλεια είναι ζωτικής σημασίας – για αυτό δεν υπάρχει αμφιβολία. Ένα μικρό παράπτωμα ασφάλειας μπορεί στην καλύτερη των περιπτώσεων να καταστήσει το site μας μη-διαθέσιμο και να χάσουμε πολύτιμες εισπράξεις μέχρι να μας κοστίσει ολόκληρη μας την υπόληψη. Δεν προκαλεί εντύπωση λοιπόν πως η μέριμνα για την ασφάλεια είναι ένας από τους πιο βασικούς παράγοντες για πολλές επιχειρήσεις όταν σκέφτονται την ανάπτυξη κάποιου υπολογιστικού νέφους. Στην πραγματικότητα τα ρίσκα είναι τα ίδια με αυτά που αντιμετωπίζουν οι παραδοσιακές πληροφοριακές λύσεις, με την βασική διαφορά να εστιάζεται στο ότι όταν εργαζόμαστε στο νέφος η ασφάλεια δεν εναπόκειται αποκλειστικά στους δικούς μας ώμους αλλά είναι αντ' αυτού μια κοινή ευθύνη με τον πάροχο που φιλοξενεί το νέφος καθαυτό. Ένας καλός πάροχος θα έχει πολλά αντίμετρα ασφαλείας από dedicated firewalls, εξελιγμένο encryption μέχρι συστήματα ανίχνευσης εισβολών και data centers τα οποία θα είναι σύμφωνα με τα τελευταία PCI DDS, ISO και ISAE standards.

2.4.3 Είναι πάντα φτηνότερα να τρέχεις στο Νέφος

Η αλήθεια είναι πως δεν είναι πάντα φτηνότερα να έχεις ένα νέφος, αλλά μπορεί συχνά να αποδειχθεί πιο αποδοτικό όσον αφορά στο κόστος. Αν χρειαζόμαστε όλους μας τους servers να τρέχουν 24/7/365, τότε πολύ πιθανώς να μπορούμε να βρούμε την ίδια υπολογιστική ισχύ με λιγότερες δαπάνες αν χρησιμοποιήσουμε έναν dedicated server. Το νέφος λειτουργεί καλύτερα για ποικίλες απαιτήσεις και φόρτους εργασίας, όπου ορισμένες φορές υπάρχουν υψηλές απαιτήσεις και άλλες χαμηλές. Το νέφος επιτρέπει να απενεργοποιήσουμε servers αυτές τις περιόδους που η ζήτηση είναι χαμηλή για να περιοριστούν έτσι τα κόστη αδρανούς υποδομής. Με αυτό τον τρόπο μπορεί να βελτιωθεί η αποδοτικότητα του κόστους.

2.4.4 Όλα μπορούν να αυτοματοποιηθούν στο νέφος, έτσι δεν υπάρχει ανάγκη για υποστήριξη

Φυσικά, η ομορφιά του νέφους είναι πως τα πάντα από το επίπεδο υποδομής και πάνω μπορούν να αυτοματοποιηθούν, αλλά αυτό θέλει ένα σημαντικό επίπεδο τεχνογνωσίας και κατανόησης των εργαλείων που

εμπλέκονται. Για έναν προγραμματιστή εφαρμογών που γράφει εφαρμογές για να τρέξουν εξ' αρχής στο νέφος, η αυτοματοποίηση θα είναι εύκολη, με το να επιτρέψει τις δυνατότητες που αναφέραμε παραπάνω όπως η αυτόματη κλιμάκωση και η αυτόματη ανάκαμψη, χωρίς καθόλου ανθρώπινη παρέμβαση. Ωστόσο, αν δεν έχουμε αυτό το επίπεδο γνώσης τότε πολύ πιθανά θα χρειαστούμε επιπλέον υποστήριξη τουλάχιστον βραχυπρόθεσμα.

2.4.5 Το Νέφος είναι επιβλαβές για το περιβάλλον

Στην πραγματικότητα το υπολογιστικό νέφος είναι η περιβαλλοντική προσέγγιση. Τα data center χρησιμοποιούν δύο φορές περισσότερη ενέργεια απ' ότι χρειάζεται στην πραγματικότητα λαμβάνοντας ως μέρη της εξίσωσης την ψύξη, την ηλεκτροδότηση και την συντήρηση.

2.4.6 Το νέφος στερεί δουλειές

Ο μύθος που επικρατεί την επίπτωση του νέφους στην Πληροφοριακή υποδομή μπορεί εύκολα να καταρριφθεί. Υπολογίζεται ότι μέχρι το 2015 η τεχνολογία υπολογιστικού νέφους θα έχει δημιουργήσει περισσότερες από 13 εκατομμύρια δουλειές παγκοσμίως. Όπως με κάθε πρωτοποριακή τεχνολογική λύση, η κλίση του κόσμου προς το νέφος (ως τεχνολογία) θα απαιτήσει την πρόσληψη ειδικών των οποίων οι δεξιότητες και οι γνώσεις θα συντηρούν και θα προωθούν την ανάπτυξη και την ενδυνάμωση των υπολογιστικών νεφών.

2.4.7 Η μετακίνηση στο Νέφος είναι πιο επιζήμια απ' ότι η αξία της

Παρότι ίσως μπορεί να χρειαστεί κάποια διαχείριση όσον αφορά στην αρχιτεκτονική, η μεταφορά μπορεί να γίνει σχετικά ανώδυνα σε συνεργασία με έναν έμπιστο και πεπειραμένο πάροχο, ειδικά αν χρησιμοποιούμε πολύ παλιούς servers. Υπό ιδανικές συνθήκες μπορούν όλα να γίνουν πολύ γρήγορα και φαινομενικά με μηδαμινό downtime (χρόνος μη διαθεσιμότητας). Σίγουρα, η βραχυπρόθεσμη δυσκολία, δεν υπερτερεί των μακροπρόθεσμων οφελών της μεγαλύτερης αποδοτικότητας, εξοικονόμησης κόστους και μιας επιχείρησης που είναι μελλοντικά εξασφαλισμένη ανεξαρτήτως αλλαγών στην αγορά και στις απαιτήσεις.

2.4.8 Τα «Big Data» δεν είναι μείζονος σημασίας.

Αυτό είναι λάθος. Με την αύξουσα ποσότητα ψηφιακών δεδομένων, οι επιχειρήσεις παράγουν και αποθηκεύουν περισσότερες πληροφορίες από ποτέ. Αν δεν ληφθεί αυτό υπόψη θα μπορούσε να οδηγήσει σε παράλυτα και μη αποδοτικά συστήματα αποθήκευσης. Με την απαξίωση των big data από την επιχείρηση, μπορεί να διατρέξετε τον κίνδυνο να χάσετε πολύτιμες πληροφορίες και συνδέσεις απλά και μόνο λόγω μη δομημένων δεδομένων. Με την μετακίνηση στο νέφος επιτυγχάνεται μεγαλύτερη αποθηκευτική αποδοτικότητα και μια δομημένη και οργανωμένη προσέγγιση στην διαχείριση των δεδομένων.

2.4.9 Η τεχνολογία Νέφους είναι ακόμα στα σπάργανα.

Μια πρόσφατη μελέτη από την ISACA έδειξε πως το υπολογιστικό νέφος προσεγγίζει γρήγορα την ωριμότητα. Μέσα στα επόμενα τέσσερα χρόνια μπορούμε να περιμένουμε να δούμε συστηματική καινοτομία σε έναν αύξοντα ρυθμό που θα εξασφαλίσει ότι το υπολογιστικό νέφος θα καλύπτει τις ανάγκες οπουδήποτε τύπου και μεγέθους επιχείρησης/οργανισμού.

2.5 Η Ασφάλεια στο Υπολογιστικό Νέφος

Η ασφάλεια είναι ένα κύριο μέλημα όταν εμπιστευόμαστε τις κρίσιμες πληροφορίες ενός οργανισμού σε γεωγραφικά διάσπαρτες πλατφόρμες υπολογιστικού νέφους και όχι κάτω από τον άμεσο έλεγχο του οργανισμού. Επιπλέον από τις συμβατικές διαδικασίες πληροφοριακών συστημάτων ασφαλείας, ο σχεδιασμός ασφάλειας μέσα στο λογισμικό του νέφους κατά την διάρκεια της ανάπτυξης αυτού του λογισμικού μπορεί να μειώσει δραματικά την πιθανότητα επίθεσης.

Με το υπολογιστικό νέφος να παρέχει το μοντέλο SaaS, η ασφάλεια του λογισμικού είναι σημαντικό ζήτημα. Από την οπτική γωνία του χρήστη ενός υπολογιστικού νέφους, η χρήση του SaaS στο νέφος μειώνει την ανάγκη για ασφαλή ανάπτυξη λογισμικού από τον χρήστη. Η απαίτηση για ασφάλεια στην ανάπτυξη λογισμικού έγκειται πλέον στον πάροχο. Παρόλα αυτά ο χρήστης μπορεί να εξακολουθεί να βρίσκει απαραίτητο την ανάπτυξη δικού του κώδικα για το νέφος. Οποιοσδήποτε και αν αναπτύσσει λογισμικό, αυτή η διεργασία απαιτεί δυνατή αφοσίωση σε έναν επίσημο κύκλο ζωής που διέπεται από προδιαγραφές

ασφαλείας, συμπεριλαμβανομένης της σχεδίασης, του ελέγχου, της ασφαλούς ανάπτυξης, και της κατάργησης. Πάραυτα, σε πολλές περιπτώσεις, η ασφάλεια λογισμικού δεν είναι παρά μόνο ένα επιπρόσθετο στο υπάρχον λογισμικό και ένα όχι τόσο σημαντικό στοιχείο της διαδικασίας προγραμματισμού.

Αυτά και άλλα συσχετιζόμενα θέματα στον κύκλο ζωής της ασφαλούς ανάπτυξης εφαρμογών θα συζητηθούν λεπτομερώς σε αυτό το υποκεφάλαιο.

2.5.1 Στόχοι ασφαλείας του υπολογιστικού νέφους

Η Software Security Assurance Report ορίζει την **διασφάλιση λογισμικού**[15] ως την βάση για τη λήψη αποδεδειγμένης εμπιστοσύνης πως το λογισμικό θα αναδεικνύει συνεχώς όλες τις ιδιότητες που απαιτούνται για να διασφαλιστεί ότι το εκτελούμενο λογισμικό, θα συνεχίσει να λειτουργεί αξιόπιστα ανεξάρτητα από την παρουσία ελαττωμάτων. Με πιο πρακτικούς όρους, ένα τέτοιο λογισμικό πρέπει να είναι δυνατό να αντιστέκεται στα περισσότερα ήδη επιθέσεων, να αντέξει όσο το δυνατόν περισσότερες επιθέσεις από αυτές στις οποίες δεν μπορεί να αντισταθεί, και να περιορίζει την ζημία καθώς επίσης και να ανακάμπτει σε ένα φυσιολογικό επίπεδο λειτουργίας όσο το δυνατόν πιο γρήγορα μετά από μια επίθεση που δεν θα μπορέσει να αντέξει.

Το Data and Analysis Center for Software (DACS) απαιτεί το λογισμικό να παρουσιάζει τις ακόλουθες τρεις ιδιότητες για να χαρακτηριστεί ασφαλές:

- **Dependability** – Λογισμικό που εκτελείται όπως έχει προβλεφθεί και λειτουργεί σωστά κάτω από μία ποικιλία συνθηκών, συμπεριλαμβανομένου μιας επίθεσης ή το να εκτελεστεί πάνω σε κακόβουλο host.
- **Trustworthiness** – Λογισμικό που περιέχει έναν ελάχιστο αριθμό ευπαθειών ή καθόλου ευπάθειες ή αδυναμίες που θα μπορούσαν να διακυβεύσουν το dependability του λογισμικού. Πρέπει επίσης να αντιστέκεται σε κακόβουλη λογική.
- **Survivability (Resilience)** – Λογισμικό που αντιστέκεται ή είναι ανεκτικό σε επιθέσεις έχει την ικανότητα να ανακάμπτει όσο το δυνατόν πιο γρήγορα με την μικρότερη δυνατή ζημία.

Συνοδευτικές θεμελιώδεις αρχές ασφάλειας τις οποίες δεν πρέπει να ξεχνάμε αποτελούν επίσης οι εμπιστευτικότητα, ακεραιότητα και η διαθεσιμότητα οι οποίες πρέπει επίσης να μην παραβιάζονται.

2.5.2 Υπηρεσίες Ασφάλειας στο Νέφος

Το νέφος ως τεχνολογική καινοτομία πρέπει να παρέχει υπηρεσίες ασφαλείας προς τους χρήστες του για να θεωρείται αξιόπιστο. Μέσα σε αυτή τη λογική βασίζεται φυσικά και η επιχειρηματική λογική της απώλειας δεδομένων και άρα κατ' επέκταση φήμης και εσόδων για την οποιαδήποτε επιχείρηση/οργανισμό. Οι βασικές υπηρεσίες ασφάλειας που προσφέρονται σε ένα υπολογιστικό νέφος συνοψίζονται παρακάτω.

2.5.2.1 Αυθεντικοποίηση

Η αυθεντικοποίηση είναι ο έλεγχος των στοιχείων της ταυτότητας ενός χρήστη. Εδραιώνει την ταυτότητα του χρήστη στο σύστημα και διασφαλίζει ότι οι χρήστες είναι αυτοί που ισχυρίζονται. Για παράδειγμα, ένας χρήστης παρουσιάζει το αναγνωριστικό του σε μια οθόνη εισόδου και ακολούθως τον κωδικό του. Το υπολογιστικό σύστημα (ή στην περίπτωση μας το νέφος) αυθεντικοποιεί τον χρήστη με το να επικυρώσει ότι ο κωδικός αντιστοιχεί στο αντίστοιχο αναγνωριστικό.

2.5.2.2 Εξουσιοδότηση

Η εξουσιοδότηση αναφέρεται στα δικαιώματα και τα προνόμια που δίνονται σε ένα άτομο ή μια διεργασία που θα του επιτρέψουν την πρόσβαση στους πόρους ενός υπολογιστικού νέφους και στα σύνολα δεδομένων που αυτοί φιλοξενούν. Μόλις πραγματοποιηθεί η αυθεντικοποίηση ενός χρήστη, τα επίπεδα εξουσιοδότησης ορίζουν το μέγεθος των δικαιωμάτων που μπορεί να έχει ο χρήστης αυτός στο σύστημα.

2.5.2.3 Καταγραφή (Auditing)

Για να διατηρηθεί η ομαλή λειτουργία, χρησιμοποιούνται δύο βασικές μέθοδοι: καταγραφή συστήματος και παρακολούθηση. Αυτές οι μέθοδοι μπορούν να υιοθετηθούν από τον πελάτη του νέφους, τον πάροχο του νέφους ή και του δύο, ανάλογα την αρχιτεκτονική και την ανάπτυξη.

- Μια καταγραφή συστήματος είναι ένα μοναδικό ή περιοδικό γεγονός που αξιολογεί την ασφάλεια.
- Η παρακολούθηση αναφέρεται σε μια εκτελούμενη δραστηριότητα η οποία ελέγχει είτε το σύστημα είτε τους χρήστες, όπως για παράδειγμα ανίχνευση εισβολών.

Οι καταγραφείς συστημάτων πληροφορικής συνήθως καταγράφουν τις ακόλουθες λειτουργίες:

- Ελέγχους συστήματος και συναλλαγών.
- Επίπεδα συστημάτων ανάπτυξης.
- Ελέγχους αντιγράφων.
- Διεργασίες βιβλιοθηκών δεδομένων.
- Ασφάλεια κέντρου δεδομένων (data center).
- Σχέδια επείγουσας επέμβασης.

Ένα *ίχνος καταγραφής* γνωστό και ως log είναι ένα πακέτο εγγραφών που παρέχει στοιχεία διεργασιών που εκτελέσθηκαν, και χρησιμοποιείται για να ελέγξει τις κατευθύνσεις των συναλλαγών δεδομένων αυτών των διεργασιών. Τα ίχνη καταγραφής μπορεί να περιοριστούν σε συγκεκριμένα γεγονότα ή μπορεί να εποπτεύουν όλες τις διεργασίες σε ένα υπολογιστικό νέφος.

Τα ίχνη θα πρέπει να καταγράφουν τα ακόλουθα:

- Ημερομηνία και ώρα συναλλαγής.
- Ποια διεργασία έκανε τη συναλλαγή.
- Σε ποιο τερματικό έγινε η συναλλαγή.
- Διάφορα γεγονότα ασφαλείας συσχετιζόμενα με την συναλλαγή.

Με τον όρο συναλλαγή αναφερόμαστε πάντα σε μεταφορά (ή/και ανταλλαγή) δεδομένων μεταξύ τερματικών.

2.5.2.4 Ευθύνη (Accountability)

Ευθύνη είναι η δυνατότητα να καθοριστούν οι δράσεις και οι συμπεριφορές ενός ατόμου μέσα σε ένα σύστημα υπολογιστικού νέφους για να αναγνωριστεί το συγκεκριμένο άτομο. Τα ίχνη καταγραφής και τα logs υποστηρίζουν αυτή την Ευθύνη και μπορεί να χρησιμοποιηθούν για να συσταθούν μεταμοντέρνες μελέτες προκειμένου να αναλυθούν ιστορικά γεγονότα και τα άτομα ή τις διεργασίες που συσχετίζονται με αυτά τα γεγονότα. Η Ευθύνη συσχετίζεται με την έννοια της «μη αποκήρυξης» όπου ένα άτομο δεν μπορεί επιτυχώς να αρνηθεί την εκτέλεση μιας δράσης στο σύστημα.

Επίλογος

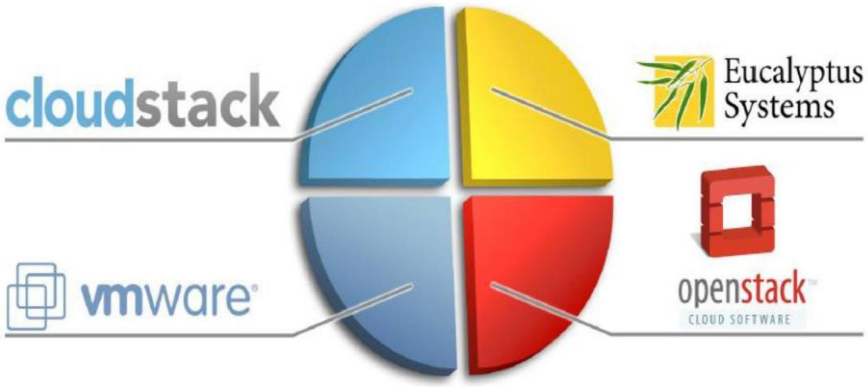
Σε αυτό το κεφάλαιο μελετήσαμε και πήραμε μια γενική ιδέα όλου του φάσματος της τεχνολογικής καινοτομίας του υπολογιστικού νέφους. Είδαμε πως μπορούν να κατηγοριοποιηθούν τα νέφη καθώς και τα χαρακτηριστικά που πρέπει να πληρούν ώστε να θεωρούνται ασφαλή. Στο επόμενο κεφάλαιο θα μπούμε στα ενδότερα του νέφους και θα δούμε ποιες ήταν οι επιλογές μας στο θέμα του λογισμικού που θα χρησιμοποιούσαμε για την σχεδίαση και την υλοποίηση του συστήματος μας. Θα μελετήσουμε τα χαρακτηριστικά της κάθε επιλογής καθώς και την ασφάλεια που παρέχει η καθεμία.

3 Περιγραφή των Τεχνολογικών Λύσεων

Εισαγωγή

Σκοπός του κεφαλαίου είναι η συγκριτική αξιολόγηση των διαθέσιμων λογισμικών υπολογιστικού νέφους. Η τεχνολογία του υπολογιστικού νέφους είναι ένα από τα κύρια θέματα στην επιστήμη των υπολογιστών σήμερα. Αυτό συμβαίνει διότι είναι πολλά υποσχόμενο ως προς την βελτίωση της απόδοσης, την μείωση του κόστους, την επεκτασιμότητα της υποδομής μας , την υψηλή απόδοση και την ασφαλή αποθήκευση δεδομένων.

Η επιλογή της καταλληλότερης πλατφόρμας νέφους είναι δύσκολή. Η κάθε μία έχει τα δικά της πλεονεκτήματα και μειονεκτήματα γι' αυτόν ακριβώς τον λόγο αποφασίσαμε να κάνουμε μια βαθύτερη έρευνα στα πιο αξιοσημείωτα συστήματα που διατίθενται, να συγκρίνουμε τις δυνατότητες τους και να συνοψίσουμε τα χαρακτηριστικά τους σε έναν πίνακα. Οι πλατφόρμες που θα πάρουν μέρος στην σύγκριση είναι το CloudStack[16], Eucalyptus[17], VCloud Director[18] και Openstack.



3.1 Τα Λογισμικά υπολογιστικού νέφους που θα μελετήσουμε

3.1 Χαρακτηριστικά των διαθέσιμων επιλογών

3.1.1 CloudStack

Το Cloudstack είναι μια κονσόλα διαχείρισης υπολογιστικών πόρων σε data centers. Πολλές γνωστές εταιρίες διαχείρισης δεδομένων όπως η Zynga, η Nokia Research Center και η Cloud Central έχουν αναπτύξει τα συστήματα νέφους τους βασισμένες στο Cloudstack. Εκτός του ότι το Cloudstack έχει αναπτύξει το δικό του API, η πλατφόρμα υποστηρίζει και το CloudBridge Amazon EC2, το οποίο ενεργοποιεί την δυνατότητα μετατροπής ενός Amazon API σε CloudStack API.

Κύρια Χαρακτηριστικά :

- ✓ Υποστήριξη πολλών Hypervisors(KVM, XEN, ESXi, OVM και BareMetal)
- ✓ Ρόλοι (Ανάθεση και διαχείριση δικαιωμάτων)
- ✓ Εικονικά Τοπικά Δίκτυα (VLAN Support)
- ✓ Resource Pool (Δίνει την δυνατότητα στους διαχειριστές να περιορίζουν τους εικονικούς πόρους του συστήματος. Για παράδειγμα, τον αριθμό εικονικών μηχανών ανά λογαριασμό χρήστη και τον αριθμό δημόσιων IP διευθύνσεων που μπορεί να δώσει σε αυτές).
- ✓ Snapshots και Volumes
- ✓ Εικονικοί δρομολογητές, Τοίχοι Προστασίας και ισορροπία του φόρτου εργασίας

Ακόμα, το CloudStack παρέχει και την κατάλληλη υποστήριξη για την ανάπτυξη του με έγγραφα και οδηγούς για τους χρήστες. Έτσι έχοντας ένα βασικό υπόβαθρο θα είναι πολύ εύκολο για κάποιον να εγκαταστήσει την πλατφόρμα του CloudStack. Στο σημείο αυτό παρουσιάζεται και ένα αρνητικό χαρακτηριστικό καθώς σε πιο περίπλοκες εφαρμογές χρήσης του CloudStack η βιβλιογραφία του δεν τις καλύπτει. Οι οδηγοί μπορεί να δίνουν συμβουλές, βήμα-βήμα, αλλά δεν παρέχουν πληροφορίες για το πώς λειτουργεί η πλατφόρμα σε γενικό πλαίσιο.

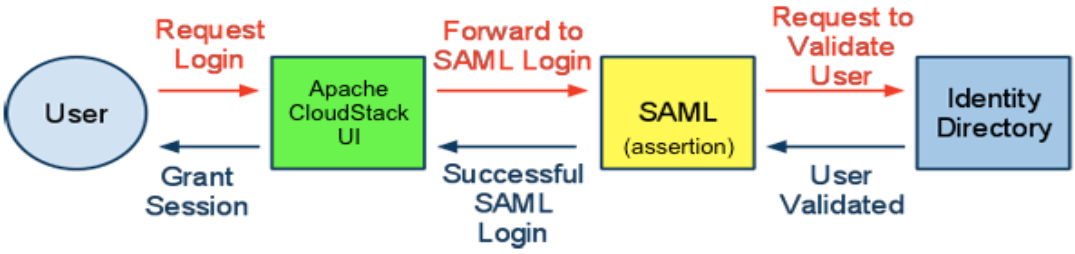
Στην συνέχεια, όσον αφορά την κοινότητα του CloudStack, υπάρχει η δυνατότητα παροχής δωρεάν τεχνικής υποστήριξης μέσω διαδικτύου όπως επίσης εύρεση λύσεων σε τυχόν προβλήματα του CloudStack στο Forum. Στα πλαίσια της κοινότητας και παροχής βοήθειας υπάρχει η δυνατότητα επικοινωνίας μέσω IRC καναλιού για οποιαδήποτε ερώτηση έχει κάποιος χρήστης.

Τέλος, το CloudStack κατά την χρήση του μπορεί να παρουσιάσει πολλά bugs, άλλα από αυτά μπορεί να είναι γνωστά προς τους χρήστες του και άλλα μπορεί να εμφανίζονται για πρώτη φορά.

Σε γενικές γραμμές το CloudStack αφήνει θετικές εντυπώσεις καθώς αποτελεί ένα πολλά υποσχόμενο εργαλείο το οποίο αναπτύσσεται με γοργούς ρυθμούς και παρέχει ευρεία λειτουργικότητα και ασφαλώς είναι δωρεάν προς χρήση.

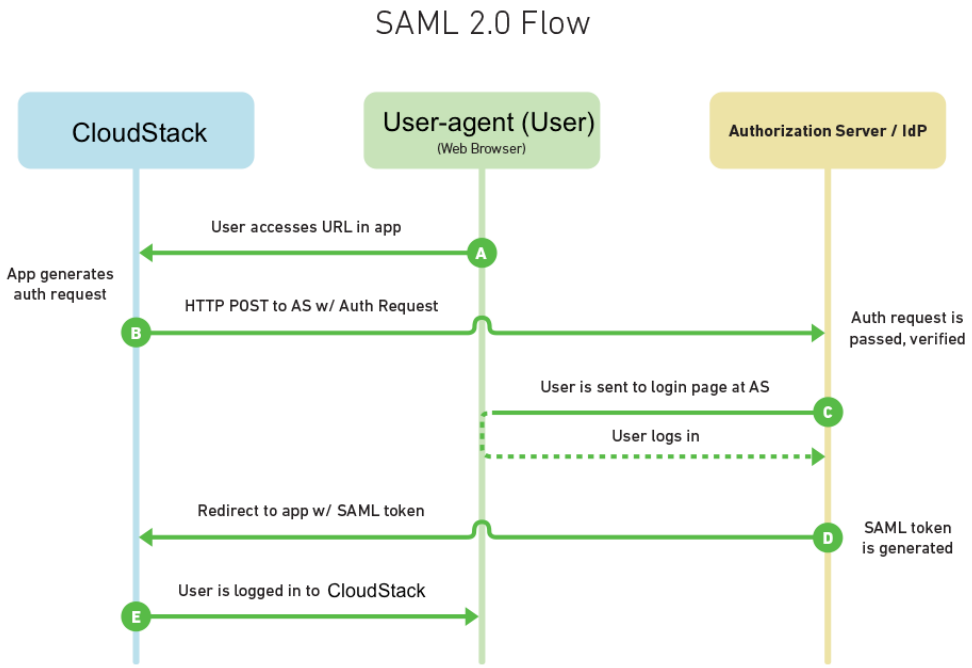
3.1.1.1 Η ασφάλεια του CloudStack

Ως προς την ασφάλεια του το Cloudstack χρησιμοποιεί τον δικό του μηχανισμό αυθεντικοποίησης. Υπάρχουν δύο μέθοδοι στον μηχανισμό αυτόν. Η πρώτη μέθοδος περιλαμβάνει την χρησιμοποίηση του ονόματος και του κωδικού χρήστη και είναι βασισμένη σε ένα cookie και σε ένα κλειδί συνεδρίασης(session-key). Αντίστοιχα, η δεύτερη μέθοδος είναι βασισμένη σε μία υπογραφή HMAC και χρησιμοποιεί ένα API[19] και ένα μυστικό κλειδί. Πολλοί οργανισμοί θα επιθυμούσαν να χρησιμοποιήσουν αυτόν τον μηχανισμό αυθεντικοποίησης που παρέχει **Single Sign On (SSO)[20]** και **Single Log Out (SLO)[21]** στο CloudStack UI και στους clients. Επίσης, χρησιμοποιείται και το **SAML(Security Assertion Markup Language)**[22]. Το SAML είναι ένα παλιό, σταθερό και διαδεδομένο πρωτόκολλο αυθεντικοποίησης και εξουσιοδότησης βασισμένο στην XML[23]. Σκοπός είναι η ενσωμάτωση του SSO SAML και η υποστήριξη του στο CloudStack. Το χαρακτηριστικό αυτό θα είναι χρήσιμο σε χρήστες που θα επιθυμούν την επαναχρησιμοποίηση μιας ήδη υπάρχουσας SAML 2.0 IdP (Identity Provider) υπηρεσίας. Αυτή η υπηρεσία έχει την ευθύνη της διαχείρισης των χρηστών, καθώς και την αυθεντικοποίηση και την εξουσιοδότηση τους. Σύμφωνα με την ορολογία του SAML, το CloudStack αποτελεί έναν **Πάροχο Υπηρεσίας (Service Provider)**. Αυτό σημαίνει ότι αποτελεί την υπηρεσία στην όποια οι χρήστες θέλουν να έχουν πρόσβαση. Έτσι ο οργανισμός θα πρέπει να έχει τον δικό του **Identity Provider (IdP)**. Το IdP είναι υπηρεσία αυθεντικοποίησης και εξουσιοδότησης και κατέχει μια λίστα των χρηστών. Τέτοιες υπηρεσίες είναι το LDAP[24], το Active Directory[25] κτλ.



3.2 Αυθεντικοποίηση με SAML

Μια SSO δομή βασισμένη στο SAML αποτελείται από τρεις οντότητες. Αυτές είναι ο **User-Agent(UA)**, ο **Service Provider(SP)** και ο **Identity Provider(IdP).** Ο UA είναι ο χρήστης, ο SP είναι η εφαρμογή στην οποία ο UA έχει πρόσβαση (το Apache CloudStack UI) και τέλος ο IdP είναι η υπηρεσία ταυτοποίησης που έχει την αρμοδιότητα της αυθεντικοποίησης ,εξουσιοδότησης και της διαχείρισης των χρηστών.



3.3 Διάγραμμα ροής του SAML

Παραπάνω στην εικόνα φαίνεται το διάγραμμα ροής του SAML.Ακολουθούν αριθμημένα βήματα που θα μας δώσουν να καταλάβουμε την λειτουργία του.

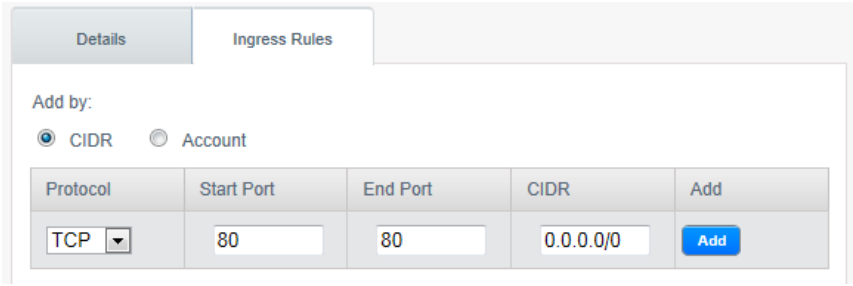
1. Αρχικά ο χρήστης επισκέπτεται το CloudStack UI και επιλέγει ένα SAML SSO κουμπί που θα τον οδηγήσει σε ένα SAML SSO URL: /client/api?command=samlsso
2. Ο χρήστης οδηγείται στο ήδη εγκατεστημένο IdP και καταχωρεί το username και το password του
3. Το IdP ελέγχει και στέλνει ένα XML response πίσω στο CloudStack SP.
4. Η εφαρμογή του SP εντός του CloudStack ελέγχει με την σειρά της το XML response, ελέγχει την υπογραφή που φέρει και αν όλα είναι σωστά συνεχίζει αλλιώς προβάλλει ένα μήνυμα αποτυχίας
5. Στην συνέχεια βρίσκει ένα NameID (identifier) το οποίο θα πρέπει να είναι μόνιμο όπως για παράδειγμα μια διεύθυνση ηλεκτρονικού ταχυδρομείου
6. Αυτό το NameID αντιμετωπίζεται σαν UUID και στην συνέχεια θα βρει έναν χρήστη ή θα δημιουργήσει έναν που θα έχει βασικό ρόλο
7. Τέλος, θα επιτρέψει στον χρήστη να εισέλθει στο CloudStack

Ένα άλλο χαρακτηριστικό του CloudStack, αυτή την φορά για την προστασία των εικονικών μηχανών του, είναι τα **Security Groups**. Τα Security groups είναι ένας τρόπους έτσι ώστε να απομονώνεται η κίνηση προς τα VMs. Είναι μια ομάδα από VMs η οποία φιλτράρει την εισερχόμενη και την εξερχόμενη κίνηση σύμφωνα με κανόνες που λέγονται **ingress** και **egress**. Αυτοί οι κανόνες φιλτράρουν την δικτυακή κίνηση σύμφωνα με την IP διεύθυνση που προσπαθεί να επικοινωνήσει με το εκάστοτε VM. Τα security groups έχουν μεγάλη χρησιμότητα σε περιπτώσεις όπου υπάρχει μια βασική δικτυακή δομή διότι υπάρχει ένα μόνο guest δίκτυο για όλα τα guest VMs. Βέβαια, σε ένα πιο πολύπλοκο δίκτυο μπορούμε να ορίσουμε πολλά guest δίκτυα για την απομόνωση της κίνησης.

Αρχικά, κάθε CloudStack λογαριασμός χρήστη, ανήκει στο βασικό Security group που αποτρέπει όλη την εισερχόμενη κίνηση και επιτρέπει όλη την εξερχόμενη. Το βασικό Security Group μπορεί να παραμετροποιηθεί ανάλογα με τις ανάγκες του κάθε συστήματος.

Το CloudStack UI μας δίνει την δυνατότητα όπως είναι λογικό να μπορούμε να διαχειριζόμαστε τα Security Groups. Μπορούμε να δημιουργούμε, να

διαγράφουμε και να τροποποιούμε ήδη υπάρχοντα groups και αντίστοιχα να μπορούμε να κάνουμε τις ίδιες ενέργειες για τους κανόνες εντός του. Στην εικόνα 3.4 που ακολουθεί μπορούμε να δούμε πως δημιουργείται ένας κανόνας εσωτερικής πρόσβασης HTTP ανεξάρτητα από την προέλευση του.



3.4 Ένας κανόνας εσωτερικής http πρόσβασης

Για την κρυπτογράφηση των δεδομένων, το CloudStack χρησιμοποιεί τους αλγόριθμους RSA, AES και την κρυπτογραφική συνάρτηση κατακερματισμού MD5.

Ο **RSA(Rivest, Shamir and Adleman)** χρησιμοποιείται για την ασύμμετρη κρυπτογράφηση δημοσίου κλειδιού. Χρησιμοποιεί δύο κλειδιά, ένα δημόσιο και ένα ιδιωτικό. Το δημόσιο κλειδί χρησιμοποιείται για την κρυπτογράφηση του μηνύματος και είναι γνωστό σε όλους. Τα μηνύματα αυτά που κρυπτογραφούνται με το δημόσιο κλειδί μπορούν να αποκρυπτογραφηθούν μόνο με την χρήση του ιδιωτικού κλειδιού. Στην περίπτωση της αυθεντικοποίησης του με τον server, ο server εφαρμόζει αυθεντικοποίηση δημοσίου κλειδιού υπογράφοντας ένα μήνυμα με το ιδιωτικό κλειδί του, δημιουργώντας έτσι αυτό που ονομάζεται ψηφιακή υπογραφή. Η υπογραφή αυτή στην συνέχεια επιστρέφεται στον χρήστη, ο οποίος την επαληθεύει χρησιμοποιώντας το γνωστό δημόσιο κλειδί του server.

Ο **AES (Advanced Encryption Standard)** είναι ένας κωδικοποιητής τμημάτων με συμμετρικό σύστημα κρυπτογράφησης, με μήκος τμήματος 128 bit και υποστηρίζει κλειδιά μήκους 128-bit, 192-bit και 256-bit. Στον αλγόριθμο AES υπάρχουν 10-15 γύροι ανάλογα με το μήκος του κλειδιού ώστε να γίνει η κρυπτογράφηση. Κάθε γύρος έχει 4 ξεχωριστά βήματα:

- Αντικατάσταση Byte(Byte Substitution)
- Ολίσθηση (Shift Row)
- Συνδυασμός Πολλών bit (Mix Column)
- Πρόσθεση του κλειδιού (Add Round key)

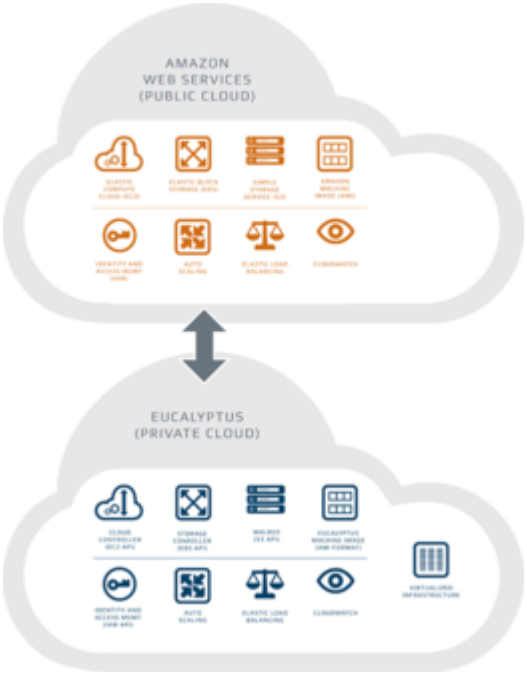
Αντίστοιχα, για την αποκρυπτογράφηση, η διαδικασία είναι ή ίδια απλά τα βήματα γίνονται με την αντίθετη σειρά.

Ο αλγόριθμος **MD5(Message Digest)** αναπτύχθηκε από τον R.Rivest και περιγράφεται στο RFC 1321. Ο αλγόριθμος λαμβάνει ως είσοδο μήνυμα αυθαίρετου μήκους και παράγει ως έξοδο μια σύνοψη των 128 bits. Η είσοδος χωρίζεται σε τμήματα των 512 bits για να επεξεργαστεί. Ο αποστολέας των δεδομένων χρησιμοποιεί το δημόσιο κλειδί του για να κρυπτογραφήσει το μήνυμα και ο παραλήπτης αντίστοιχα το ιδιωτικό του για την αποκρυπτογράφηση.

3.1.2 Eucalyptus

Το Eucalyptus αποτελεί και αυτό με την σειρά του μια ευρέως διαδεδομένη πλατφόρμα νέφους. Εταιρίες και οργανισμοί όπως Sony, Puma, NASA, Trend Micro κ.α. το έχουν επιλέξει για την ανάπτυξη των ιδιωτικών νεφών τους. Το Eucalyptus διαθέτει δύο εκδόσεις. Η μία είναι δωρεάν και η δεύτερη είναι εμπορική και είναι προφανές πως διατίθεται με περισσότερα χαρακτηριστικά και λειτουργίες.

Ένα από τα χαρακτηριστικά που κάνουν την πλατφόρμα αυτή βολική στην χρήση της είναι η συμβατότητα του Eucalyptus API με το Amazon API.Αποτέλεσμα αυτού του χαρακτηριστικού είναι ότι όλα τα scripts και τα λογισμικά προϊόντα που βασίζονται στο Amazon API να μπορούν εύκολα να μεταφερθούν και να χρησιμοποιηθούν στα ιδιωτικά νέφη που δημιουργήθηκαν με το Eucalyptus. Το Eucalyptus υποστηρίζει τρεις Hypervisors XEN[26], KVM[27], ESXi[28], με τον τελευταίο να είναι διαθέσιμος μόνο στην εμπορική έκδοση.



3.5 Συμβατότητα Eucalyptus με Amazon API

Κύρια Χαρακτηριστικά :

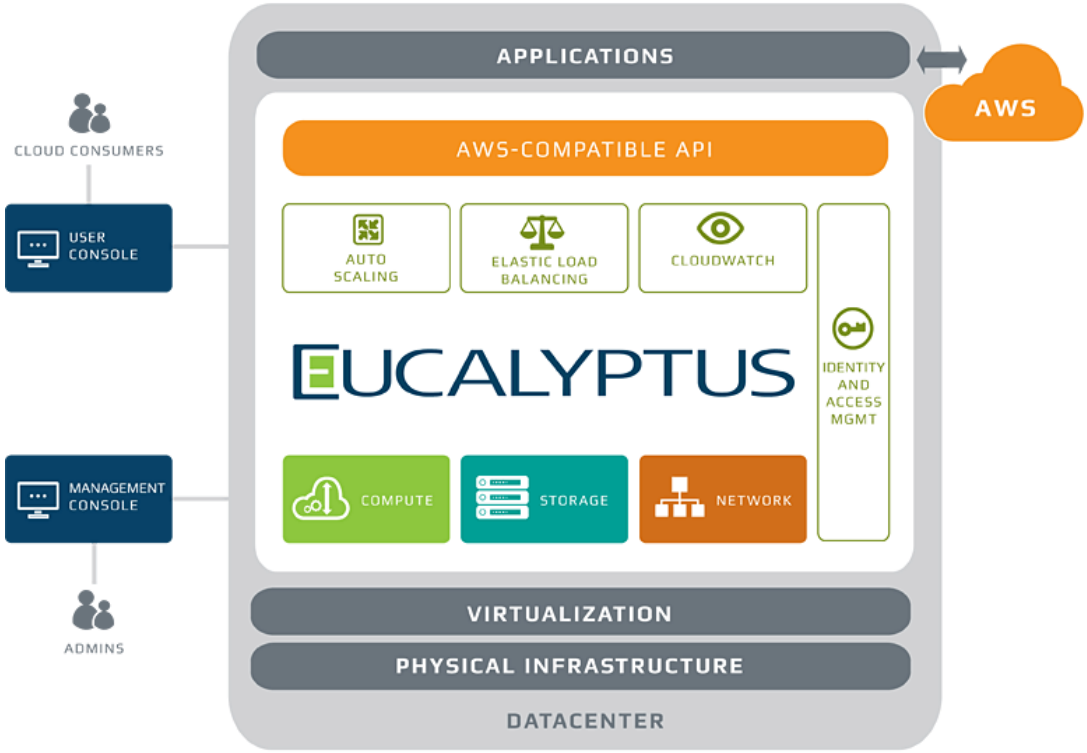
- ✓ Ρόλοι (Ανάθεση και διαχείριση δικαιωμάτων)
- ✓ Hypervisor Agnostic
- ✓ Clustering and zoning
- ✓ Ευέλικτη διαχείριση δικτύου, ομάδες ασφαλείας και απομόνωση της κυκλοφορίας

Όπως οποιοδήποτε άλλο προϊόν ανοιχτού κώδικα έτσι και το Eucalyptus διαθέτει μια ισχυρή κοινότητα η οποία προσφέρει στην ανάπτυξη της πλατφόρμας και παρέχει βοήθεια στην εύρεση και αντιμετώπιση των διαφόρων bugs που θα εμφανιστούν.

Η βιβλιογραφία που παρέχεται για το συγκεκριμένο προϊόν καλύπτει την διαδικασίας της εγκατάστασης του και δεν περιγράφεται καμία άλλη πτυχή που μπορεί το Eucalyptus να φανεί χρήσιμο, αποτελώντας ένα αρνητικό στοιχείο. Παράλληλα, ο χρήστης που θα θελήσει να ασχοληθεί με το Eucalyptus θα πρέπει να έχει πολλές τεχνικές γνώσεις καθώς οι οδηγοί που παρέχονται δεν δίνουν πληροφορίες πάνω σε θέματα virtualization με αποτέλεσμα να αχρηστεύονται όταν πρέπει να γίνει μία πιο περίπλοκη ρύθμιση.

Συνοψίζοντας, αν και η έκδοση ανοικτού κώδικα του προϊόντος έχει μια σειρά από ζητήματα, υπάρχουν απλές λύσεις για να αντιμετωπιστούν. Επιπλέον, η εμπορική έκδοση παρέχει μια ευρύτερη λειτουργικότητα (VMware Hypervisor εργαλεία, συμβατότητα με το Amazon WS[29], AD και LDAP υποστήριξη, κλπ). Η χρήση του Eucalyptus θα ήταν ιδανική για όσους έχουν ήδη ένα εικονικό περιβάλλον και επιθυμούν να το ενισχύσουν με την χρήση του Eucalyptus αντί να το αντικαταστήσουν πλήρως.

3.1.2.1 *Η ασφάλεια στο Eucalyptus*



3.6 Επισκόπηση της Αρχιτεκτονικής του Eucalyptus

Το Eucalyptus παρέχει την επιθυμητή ασφάλεια σε επίπεδο εικονικών μηχανών. Αυτό συμβαίνει με την χρησιμοποίηση διαφόρων τεχνολογιών ασφαλείας. Τέτοιες τεχνολογίες είναι:

- Firewall
- Intrusion detection και prevention systems(IDS/IPS)
- File Integrity Monitoring
- Log Inspection
- Anti-Malware Protection

Η χρήση ενός τοίχους προστασίας μειώνει την επιφάνεια επίθεσης των εικονικών servers σε περιβάλλοντα υπολογιστικού νέφους. Ένα αμφίδρομο τοίχος προστασίας, αναπτυγμένο για κάθε εικονικό μηχάνημα ξεχωριστά, μπορεί να παρέχουν κεντρική διαχείριση της πολιτικής του για τον εκάστοτε server.

Συστήματα ανίχνευσης και πρόληψης εισβολών (IDS / IPS) παρεμβαίνουν ενάντια στις επιθέσεις που επιχειρούν να εκμεταλλευτούν γνωστά τρωτά σημεία πολύ πριν την δημοσίευση και την εφαρμογή των patches που τα διορθώνουν. Εγκαθιστώντας IDS / IPS συστήματα εντός του εικονικού περιβάλλοντος μπορούμε να προστατεύσουμε τις εφαρμογές μας και τα λειτουργικά μας συστήματα από ευπάθειες που ανακαλύφθηκαν πρόσφατα. Με αυτόν τον τρόπο μπορούμε να επιτύχουμε έγκαιρη προστασία από γνωστές και “zero day” επιθέσεις.

Παρακολουθώντας την ακεραιότητα των αρχείων, ελέγχουμε τα αρχεία, τα συστήματα και το μητρώο για αλλαγές. Η παρακολούθηση της ακεραιότητας ζωτικών λειτουργικών συστημάτων και αρχείων εφαρμογών (π.χ., αρχεία, κατάλογοι, τιμές και κλειδιά μητρώου κλπ) είναι απαραίτητη για την ανίχνευση κακόβουλων και απροσδόκητων αλλαγών που θα μπορούσαν να σημάνουν την έκθεση εικονικών και υπολογιστικού νέφους πόρων σε κίνδυνο.

Η επιθεώρηση παρέχει ορατότητα σε σημαντικά γεγονότα ασφαλείας τα οποία αποθηκεύονται σε log files. Κανόνες επιθεώρησης βελτιστοποιούν τον εντοπισμό και τον προσδιορισμό των σημαντικών γεγονότων ασφαλείας τα οποία μπορεί να είναι καταχωρημένα από πολλαπλές πηγές. Αυτά τα γεγονότα μπορούν να συγκεντρωθούν και να σταλούν σε ένα αυτόνομα σύστημα ασφαλείας ή να προωθηθούν σε ένα σύστημα πληροφοριών ασφαλείας και διαχείρισης γεγονότων(Security Information Event Management – SIEM), ώστε να συσχετιστούν με άλλα γεγονότα και στην συνέχεια να αρχειοθετηθούν.

Τέλος, η Anti-Malware προστασία παρέχεται ενάντια σε ιούς, spyware, Trojans και άλλα κακόβουλα προγράμματα. Θα πρέπει να τα ανιχνεύει σε πραγματικό χρόνο και να ενσωματώνει διάφορες δυνατότητες καθαρισμού τους και να βοηθά στην αφαίρεση κακόβουλου κώδικα. Επίσης θα πρέπει να συμβάλει στην αποκατάσταση οποιασδήποτε βλάβης του συστήματος από το κακόβουλο λογισμικό.

Το Eucalyptus, όπως και το CloudStack, χρησιμοποιεί τους RSA, AES και MD5 αλγόριθμους, οι οποίοι αναφέρθηκαν παραπάνω, για την διασφάλιση των δεδομένων του υπολογιστικού νέφους.

3.1.3 vCloud Director

Το vCloud Director είναι η πλατφόρμα της VMWare που χρησιμοποιείται για την κατασκευή υπολογιστικών νεφών. Το σύστημα επιτρέπει την κατασκευή hybrid νεφών έχοντας ένα ιδιαίτερο χαρακτηριστικό. Εάν όλη η κατασκευή του νέφους έχει γίνει με την χρήση προϊόντων της VMWare, τότε δεν θα υπάρχει καμία δυσκολία στην ενσωμάτωση του vCloud Director. Δίνεται η δυνατότητα μεταφοράς και ενσωμάτωσης εικονικών μηχανών μεταξύ ιδιωτικών και δημοσίων νεφών με την χρησιμοποίηση του VMWare vCloud Connector.

Κύρια Χαρακτηριστικά :

- ✓ Εικονικά Data Centers
- ✓ vShield security technologies
- ✓ Infrastructure service catalog
- ✓ Multi-tenant organizations
- ✓ Self-service portal
- ✓ VMware vCloud API, open virtualization format, callouts

Η βιβλιογραφική υποστήριξη που συνοδεύει το λογισμικό είναι υψηλής ποιότητας. Αυτό είναι λογικό διότι το προϊόν είναι επί πληρωμή. Κατά την εγκατάσταση του προϊόντος οι οδηγίες που δίνονται είναι εύκολες και αν ακολουθηθούν προσεκτικά τότε δεν προκύπτει κάποιο πρόβλημα.

Ένα ακόμα χαρακτηριστικό το οποίο μπορεί να θεωρηθεί αρνητικό από κάποιους χρήστες είναι το γεγονός ότι χρειάζεται η Red Hat έκδοση Linux για να μπορέσει να στηθεί το vCloud. Αυτό ίσως είναι αποτρεπτικό από χρήστες που είναι εξοικειωμένοι με άλλες διανομές Linux όπως CentOS, Ubuntu, Fedora κτλ.

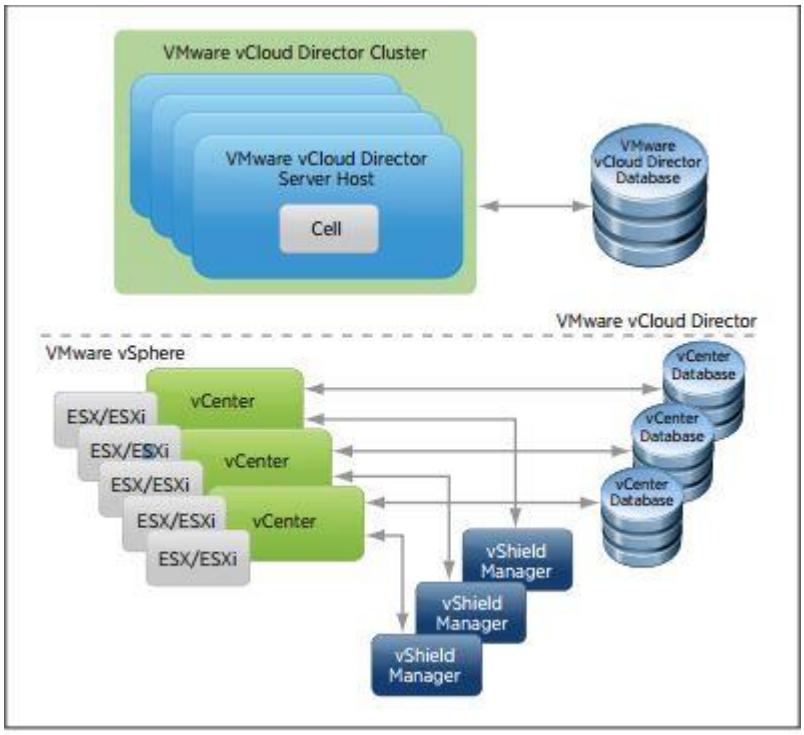
Συνοπτικά, μιας και το προϊόν αυτό είναι εμπορικό και δεν διατίθεται δωρεάν, μπορεί να αποτελέσει τροχοπέδη ώστε πολλοί ενδιαφερόμενοι να το προτιμήσουν. Βέβαια όσοι χρήστες χρησιμοποιούν προϊόντα της VMWare, το VCloud είναι σίγουρα η πιο ενδεδειγμένη λύση.

3.1.3.1 Η ασφάλεια στο vCloud Director

Στο κομμάτι της ασφάλειας το VMware vCloud Director είναι σχεδιασμένο για να δίνει περιορισμένη και ελεγχόμενη πρόσβαση στο δίκτυο, στους υπολογιστικούς και αποθηκευτικούς πόρους, για οποιονδήποτε χρήστη που χρησιμοποιεί το νέφος. Ένα από τα κύρια χαρακτηριστικά ασφαλείας του VMware vCloud Director είναι ότι δεν παρέχει άμεση ορατότητα ή πρόσβαση με πόρους επιπέδου συστήματος, συμπεριλαμβανομένου φυσικών πληροφοριών του host όπως IP διευθύνσεις, MAC διευθύνσεις, τύπος CPU, VMware ESX πρόσβαση, φυσική τοποθεσία των αποθηκευτικών μέσων κτλ. Ωστόσο, κάποιοι χρήστες μπορεί να προσπαθήσουν να αποκτήσουν πρόσβαση σε πληροφορίες σχετικά με την υποδομή του συστήματος στο οποίο τρέχουν οι ενεργοποιημένες στο νέφος εφαρμογές τους. Αν ήταν σε θέση να το πράξουν, θα μπορούσαν να ξεκινήσουν επιθέσεις στα χαμηλότερα επίπεδα του συστήματος.

Ακόμη και στο επίπεδο εικονικών πόρων, οι χρήστες μπορεί να επιχειρήσουν χρησιμοποιώντας την νόμιμη πρόσβαση τους, να αποκτήσουν μη εξουσιοδοτημένη πρόσβαση σε τμήματα του συστήματος που δεν δικαιούνται, όπως για παράδειγμα σε πόρους που ανήκουν σε άλλο οργανισμό. Μπορεί να προσπαθήσουν την απόκτηση περισσοτέρων δικαιωμάτων και προνομίων, ειδικότερα, την πρόσβαση σε δράσεις που προορίζονται για τους διαχειριστές. Χρήστες, επίσης, μπορεί να προβούν σε ενέργειες που θα διαταράξουν την συνολική διαθεσιμότητα και επίδοση του συστήματος, που σε ακραίες περιπτώσεις έχει ως αποτέλεσμα το “denial of service” άλλων χρηστών.

Η πηγή εξωτερικών κινδύνων είναι συστήματα και χρήστες εξωτερικά του νέφους, συμπεριλαμβανομένων και επιθέσεων από το Internet, εναντίον του VMware vCloud Director, μέσω του API, της Web κονσόλας (η οποία είναι γραμμένη se Adobe Flex), της vApp υπηρεσίας μεταφοράς και της απομακρυσμένης κονσόλας μιας εικονικής μηχανής. Ένας απομακρυσμένος χρήστης ο οποίος δεν έχει δικαιώματα πρόσβασης στο σύστημα μπορεί να προσπαθήσει να αποκτήσει δικαιώματα ως εξουσιοδοτημένος χρήστης. Επίσης, χρήστες που έχουν αυθεντικοποιηθεί από τα σύστημα, υπάρχει περίπτωση να είναι πιθανός κίνδυνος προς αυτό, καθώς μπορεί να προσπαθήσουν να εκμεταλλευτούν τρωτά σημεία του συστήματος τα οποία δεν είναι διαθέσιμα σε μη αυθεντικοποιημένους χρήστες.



3.7 Αρχιτεκτονική του VMWare vCloud Director

Η εικόνα παραπάνω, μας παρουσιάζει την αρχιτεκτονική του vCloud Director. Πιο συγκεκριμένα απεικονίζει ένα cluster. Εντός αυτού του cluster όπως φαίνεται μπορούν να λειτουργούν πολλαπλοί Server Hosts με τον καθένα να έχει το δικό του cell. Με όλους αυτούς τους hosts, το cluster μοιράζεται την VMWare vCloud Oracle database, καθώς και ένα Network File System(NFS), το οποίο όμως δεν απεικονίζεται στην συγκεκριμένη περίπτωση.

Όταν αναφερόμαστε στην ασφάλεια και την απομόνωση του δικτύου, σκοπός μας είναι να εκτιμήσουμε τους κινδύνους που υπάρχουν όταν ο διαχωρισμός του δικτύου και τα εργαλεία απομόνωσης της κίνησης του είναι ανεπαρκή. Επίσης , αυτοσκοπό αποτελεί η επιλογή των προτεινόμενων διορθωτικών επιλογών. Στην συγκεκριμένη περίπτωση όταν αναφερόμαστε στην κατάτμηση του δικτύου, στο μυαλό μας έρχεται μια ζώνη εμπιστοσύνης (Trust Zone).

Αυτές οι ζώνες εμπιστοσύνης είναι εργαλεία δυναμικού ελέγχου της πρόσβασης στην κυκλοφορία του δικτύου. Μια τέτοια ζώνη ορίζεται ως ένα τμήμα του δικτύου εντός του οποίου τα δεδομένα κινούνται σχετικά ελεύθερα, ενώ όταν τα δεδομένα που διακινούνται εκτός της έμπιστης ζώνης, αλλά και όταν εισέρχονται σε αυτήν, υπόκεινται σε ισχυρότερους περιορισμούς.

Παραδείγματα έμπιστων ζωνών αποτελούν :

- Αποστρατικοποιημένες ζώνες (Demilitarized zones-DMZs)
- Η βιομηχανία καρτών για πληρωμές (Payment-card industry-PCI)
- Συγκεκριμένες ζώνες ιστοσελίδων, όπως ο κατακερματισμός σύμφωνα με το τμήμα ή την λειτουργία
- Καθορισμένες ζώνες εφαρμογών, όπως για παράδειγμα οι τρεις βαθμίδες μιας εφαρμογής Web

Το vCloud Director χρησιμοποιεί ψηφιακά πιστοποιητικά για να υπάρχει ασφαλής επικοινωνία εντός του νέφους. Τα πιστοποιητικά αυτά είναι το SSL ή το TLS, τα οποία παρέχουν το απόρρητο της επικοινωνίας(με χρήση κρυπτογράφησης) και επίσης επιτρέπουν στους χρήστες να βεβαιώνουν την αυθεντικότητα του server με τον οποίο επικοινωνούν. Ο έλεγχος της ταυτότητας του server είναι αναγκαίος για να αποφευχθεί κάθε είδους man-in-the-middle επίθεση, όπου ο χρήστης επάγεται για να συνδεθεί σε έναν server που διαπράττει spoofing ή proxying για τον server που υποτίθεται πως αυτός ο χρήστης επικοινωνεί. Για την επεξεργασία αιτήσεων Web από ένα πρόγραμμα περιήγησης ή έναν REST API client, το vCloud Director υποστηρίζει την έκδοση 1.0 του TLS προτύπου(TLSv1.0), καθώς και την έκδοση 3 του παλαιότερου SSL προτύπου(SSLv3.0). Στην περίπτωση που το vCloud έχει ρόλο client, για παράδειγμα όταν επικοινωνεί με έναν Vcenter server, χρησιμοποιεί TLSv1.0 μόνο. Επίσης, το vCloud περιορίζει τα cipher suites που χρησιμοποιούνται για την κρυπτογράφηση σε εκείνα που παρέχουν ισχυρή ασφάλεια (AES και DES3).Ο έλεγχος του server εξαρτάται από την ύπαρξη ενός εγκατεστημένου πιστοποιητικού που έχει υπογραφεί από μια αναγνωρισμένη Αρχή έκδοσης Πιστοποιητικών (Certificate Authority- CA) και ταιριάζει με τον host στον οποίο ο χρήστης συνδέεται.

Η πλατφόρμα της VMWare παρέχει επίσης, υψηλή ασφάλεια σε επίπεδο δικτύου. Δικτυακά τείχη προστασίας τμηματοποιούν φυσικά ή εικονικά δίκτυα έτσι ώστε μόνο ένα περιορισμένο και σαφώς καθορισμένο σύνολο της κυκλοφορίας σε συγκεκριμένες θύρες και τα πρωτόκολλα να περάσει ανάμεσά τους. Αξίζει να σημειωθεί ότι επειδή τα cells του VMware vCloud Director είναι εντός αποστρατικοποιημένης ζώνης (DMZ), θα πρέπει να διαμεσολαβεί ένα τείχος

προστασίας ανάμεσα σε αυτά και στις διάφορες υπηρεσίες που θέλουν τα αποκτήσουν πρόσβαση.

Συγκεκριμένα, συνιστάται η χρήση τείχους για τον διαχωρισμό της DMZ από το εσωτερικό δίκτυο, σε περιπτώσεις όπως η πρόσβαση στην Oracle DB, στον vCenter Server, VSphere hosts, στον κατάλογο (LDAPv3) κτλ.

Οι εικονικές μηχανές που χρειάζονται προσβασιμότητα έξω από το νέφος (για παράδειγμα, το Internet) θα είναι είτε συνδεδεμένες σε ένα δημόσιο δίκτυο ή σε ένα ιδιωτικό NAT-routed δίκτυο με τις θύρες προώθησης διαμορφωμένες για τις κατάλληλες υπηρεσίες. Το εξωτερικό δίκτυο στο οποίο συνδέονται αυτά τα δίκτυα θα πρέπει να έχει ένα τείχος προστασίας το οποίο θα επιτρέπει, αφού πρώτα έχει συμφωνηθεί, την κίνηση προς το DMZ δίκτυο. Αυτό σημαίνει ότι, ο πάροχος των υπηρεσιών θα πρέπει να διασφαλίζει πως θα υπάρχει περιορισμός των ports και των πρωτοκόλλων που θα μπορούν να ξεκινήσουν μια σύνδεση με το εξωτερικό ή το DMZ δίκτυο, αλλά και ταυτόχρονα, θα πρέπει να διασφαλίζει την επαρκή κυκλοφορία για το δίκτυο έτσι ώστε τα vApps να μπορούν να παρέχουν τις υπηρεσίες για τις οποίες προορίζονται. Αυτό περιλαμβάνει συνήθως, την θύρα 80//TCP και 443/TCP, αλλά μερικές φορές μπορεί να περιλαμβάνει πρόσθετες θύρες και πρωτόκολλα. Ο πάροχος των υπηρεσιών αυτών, θα πρέπει να προσδιορίζει πως θα επιτυγχάνεται αυτή η ισορροπία, κατανοώντας ότι από την οπτική γωνία της ασφάλειας, περιττές θύρες πρωτόκολλα πρέπει να είναι αποκλεισμένα.

Σε γενικές γραμμές, συνιστάται τα vApps τα οποία χρειάζονται προσβασιμότητα από το Internet να τοποθετούνται σε ένα ιδιωτικό δίκτυο το οποίο έχει δρομολογηθεί. Με αυτόν τον τρόπο ο εκάστοτε οργανισμός θα έχει τον έλεγχο μέσω του τείχους προστασίας και τις προώθησης των θυρών που παρέχεται από το vShield Edge. Αυτή η διαμόρφωση δεν εξαλείφει την ανάγκη για ένα δικτυακό τείχος προστασίας το οποίο θα διαχωρίζει το εξωτερικό δίκτυο που χρησιμοποιείται από τα υπόλοιπα δίκτυα του οργανισμού. Αυτό οφείλεται στο γεγονός ότι τα δημόσια δίκτυα του οργανισμού δεν έχουν τείχος προστασίας μέσω του vShield.

Υπάρχουν μερικοί κανόνες για το τείχος προστασίας, που προτείνονται για την ασφάλεια του εσωτερικού δικτύου από δικτυακές απειλές:

1. Η απόρριψη πακέτων των οποίων η προέλευση προέρχεται από IP διευθύνεις που δεν έχουν δρομολογηθεί.
2. Η απόρριψη πακέτων που έχουν αλλάξει κατά την προώθηση τους
3. Ο περιορισμός του αριθμού των αιτήσεων, ιδιαίτερα των αιτήσεων SYN – για την προστασία απέναντι στην επίθεση της υπερχείλισης SYN αιτήσεων(προσπάθεια Denial of Service – DoS)
4. Η σκέψη για άρνηση της εξερχόμενης κίνησης από το τείχος προστασίας που δεν προέρχεται από μια εισερχόμενη αίτηση

Το VCloud Director υποστηρίζει και συνιστά την ύπαρξη τείχους προστασίας εφαρμογών Web(**Web Application Firewall – WAF**). Ένα WAF παρέχει την απαραίτητη τεχνολογία για την ασφάλεια των δεδομένων. Η τεχνολογία αυτή είναι σχεδιασμένη να προστατεύει ιστοσελίδες που φιλοξενούν εφαρμογές Web από επίδοξους εισβολείς. Λύσεις WAF είναι ικανές να εμποδίσουν τις επιθέσεις που δικτυακά τείχη προστασίας και συστήματα ανίχνευσης εισβολών δεν μπορούν. Μια λύση WAF είναι σαν ένα IDS/IPS το οποίο είναι σχεδιασμένο μόνο για την ανίχνευση και την προστασία από μια συγκεκριμένη απειλή. Με τον καθορισμό των πόρων του WAF σε δύο κυρίως πρωτόκολλα(HTTP/HTTPS), ή κάθε λύση μπορεί να αγνοεί τα πάντα εκτός από απειλές που σχετίζονται με το Web, συμπεριλαμβανομένου και επιθέσεις σε επίπεδο λειτουργικού συστήματος καθώς και τρωτά σημεία σε εφαρμογές τρίτων. Ακόμα , επειδή ένα WAF επικεντρώνεται σε ένα μόνο πρόβλημα, μπορεί να σχεδιαστεί με διάφορους τρόπους οι οποίοι δίνουν περισσότερη δύναμη και διορατικότητα σε ό, τι συμβαίνει στην πραγματικότητα στον Web server. Ως εκ τούτου, όσον αφορά την Web κυκλοφορία, η νοημοσύνη ενός τυπικού WAF είναι πολύ εξελιγμένη.

Ακολουθούν μερικά παραδείγματα κανόνων ενός WAF:

1. Η απόρριψη αιτήσεων που ενώ φαίνονται ως HTTP δεν συμβαδίζουν με τα HTTP standards όπως το RFC 2616 και το 1945
2. Ο περιορισμός του μεγέθους του HTTP σώματος και κεφαλίδων
3. Ο εντοπισμός συνηθισμένων επιθέσεων όπως π.χ. SQL Injections

Συνοψίζοντας, ένα WAF είναι μια πολύ καλή λύση ασφαλείας μια και οι εφαρμογές Web είναι πολύ εξελιγμένες ώστε ένα σύστημα IDS/IPS να τις προστατεύσει. Το απλό γεγονός ότι κάθε εφαρμογή Web είναι μοναδική καθιστά

υπερβολικά πολύπλοκη την προστασία με την χρήση μια στατικής λύσης που ταιριάζει πρότυπα.

Στην συνέχεια, αποθαρρυντικός παράγοντας για την χρησιμοποίηση του vCloud αποτελεί η οικονομική πολιτική του, καθώς δεν υπάρχουν δωρεάν εκδόσεις του. Ανάλογα με το πακέτο που επιθυμεί ο πελάτες που εξαρτάται από υπολογιστικούς πόρους, χρόνο χρησιμοποίησης κτλ, εξαρτάται και η τιμή του.

Μιας και η εταιρία της VMWare είναι από τους κολοσσούς στο θέμα των εικονικών υπολογιστικών συστημάτων και γενικά του virtualization, υπάρχει μια τεράστια κοινότητα γύρω από τα προϊόντα της. Υπάρχει επίσης, μια τεράστια βάση με ερωτήσεις και λύσεις όπου μπορεί να χρησιμοποιηθεί δωρεάν ως τεχνική υποστήριξη. Χαρακτηριστικό είναι το γεγονός ότι, η αγορά του προϊόντος συνοδεύεται από τεχνική υποστήριξη αλλά, υπάρχει και η δυνατότητα παροχής εξειδικευμένης και προσωπικής βοήθειας επί πληρωμή. Το WAF είναι μοναδικό συστατικό ασφαλείας, επειδή έχει την ικανότητα να κατανοεί ποιοι χαρακτήρες επιτρέπονται εντός του περιεχομένου των πολλών κομματιών και μερών που απαρτίζουν μια ιστοσελίδα.

Τέλος όσον αφορά την ασφάλεια συνιστάται ιδιαίτερα η ρύθμιση ενός LDAPv3 καταλόγου για την ταυτοποίηση του χρήστη. Το VMware vCloud Director πρέπει να ρυθμιστεί ώστε να γίνεται σύνδεση στους LDAP Servers μέσω SSL ώστε οι κωδικοί πρόσβασης να προστατεύονται σωστά καθώς επικυρώνονται σε σχέση με αυτούς τους servers. Για να γίνει αυτό:

- Επιλέγουμε την χρήση του SSL στις ρυθμίσεις του LDAP
- Δίνεται η δυνατότητα τις επιλογής των πιστοποιητικών, έτσι επιλέγουμε κάποιο από αυτά ή ακόμα και όλα. Εναλλακτικά, υπάρχει η επιλογή ενός KeyStore
    - Συνιστάται να μην επιλέξουμε την χρήση όλων των πιστοποιητικών. Αν επιλέξουμε όλα τα πιστοποιητικά τότε αφαιρείται η ικανότητα του συστήματος να ταυτοποιεί τον server και να διαπιστώνει αν το LDAP server έχει πλαστογραφηθεί
    - Συνιστάται η παροχή του συγκεκριμένου πιστοποιητικού του LDAP server στον οποίο γίνεται η σύνδεση. Με αυτόν τον τρόπο θα υπάρχει μεγαλύτερος έλεγχος στους servers στους οποίους

το vCloud Director θα συνδεθεί και στην εμπιστοσύνη για την ταυτοποίηση των χρηστών.

- Υπάρχει η εναλλακτική περίπτωση της χρήσης ενός JCE KeyStore για να καθοριστεί ποιο πιστοποιητικό εμπιστευόμαστε. Ωστόσο, ένα KeyStore με πολλά CA πιστοποιητικά (ή και ακόμα με πολλά, συγκεκριμένα πιστοποιητικά για servers) αποδέχεται πολλά πιστοποιητικά ως έμπιστα και αξιόπιστα.

Επιπροσθέτως, απαιτείται και συνδεσιμότητα με τον LDAP server. Ενώ το απλό (όχι SSL) LDAP τρέχει μέσω της θύρας 389/TCP, οι servers που υποστηρίζουν το LDAP μέσω SSL χρησιμοποιούν την θύρα 636/TCP με την δυνατότητα ρύθμισης της. Αξίζει να τονιστεί ότι το vCloud Director υποστηρίζει το Legacy LDAP μέσω SSL(LDAPS) και δεν υποστηρίζει την διαπραγμάτευση TLS μέσα σε μια LDAP σύνδεση, χρησιμοποιώντας την εντολή StartTLS. Τέλος, ο ενεργοποιημένος κατάλογος του LDAP server πρέπει να είναι σωστά ρυθμισμένος με ένα πιστοποιητικό SSL. Αν το συγκεκριμένο υπογεγραμμένο SSL πιστοποιητικό δεν παρέχεται απευθείας στο Web UI, τότε το πιστοποιητικό της CA που υπογράφει αυτό του LDAP server πρέπει να εισαχθεί στο JCE KeyStore. Το KeyStore το διαχειριζόμαστε με ειδικά commands που παρέχονται σε συγκεκριμένη βιβλιογραφία, ειδικά, για τον σκοπό αυτό.

3.1.4 Openstack

Το Openstack αποτελεί και αυτό, όπως το CloudStack και το Eucalyptus, λογισμικό ελεύθερου κώδικα το οποίο χρησιμοποιείται για ανάπτυξη συστημάτων νέφους. Τρεις συνιστώσες υπηρεσίες αποτελούν τον πυρήνα του, Nova, Swift και Glance. Το Openstack υποστηρίζει όλους τους Hypervisors που χρησιμοποιούνται κατά κόρον όπως οι XEN, KVM, Hyper-V, BareMetal κτλ. Είναι άξιο λόγου να αναφέρουμε ότι από όλες τις πλατφόρμες που αναφέραμε μέχρι τώρα το Openstack αποτελεί την πιο διαδεδομένη απ’ όλες αλλά και αυτή με την μεγαλύτερη και γρηγορότερη ανάπτυξη. Επίσης το Openstack είναι το πιο δημοφιλές και χρησιμοποιείται από τους ειδικούς του είδους και από εταιρίες και οργανισμούς όπως η CISCO, NASA, Dell, Intel, AMD, Rackspace, Right Scale κ.α.

Κύρια Χαρακτηριστικά :

- ✓ Διαχείριση των πόρων εικονικών servers
- ✓ Διαχείριση τοπικών δικτύων
- ✓ Διαχείρισης εικόνων με την χρήση εικονικών μηχανών
- ✓ Security Groups
- ✓ Έλεγχος πρόσβασης βασισμένος σε ρόλους
- ✓ VNC proxy μέσω ενός browser
- ✓ Projects

Όπως προαναφέρθηκε στην εισαγωγή του Openstack, αυτό διανέμεται δωρεάν, το λογισμικό αναπτύσσεται και συντηρείται από την κοινότητα του και η οικονομική ενίσχυση και ύπαρξη του οφείλεται σε δωρεές των χρηστών.

Στην συνέχεια, αναφορικά με την κοινότητα του Openstack, αυτή είναι που κρατάει ζωντανό το προϊόν αυτό. Τόσο το τεράστιο μέγεθος της, όσο και η ενεργή συμβολή της στην ανάπτυξη και στην συντήρηση του Openstack είναι τα κύρια χαρακτηριστικά που την κάνουν καλύτερη σε σχέση με τις κοινότητες των άλλων προϊόντων που είδαμε.

Ακόμα, το Openstack παρέχει την καλύτερη βιβλιογραφία και τους καλύτερους οδηγούς σε σχέση με τις άλλες πλατφόρμες. Αξίζει να σημειωθεί ότι, η εγγραφή όλων των οδηγών και κάθε άλλης έντυπης βοήθειας είναι προσφορά των χρηστών της κοινότητας. Έτσι, το Openstack δεν παρέχει απλά έναν γενικό οδηγό αλλά, παρέχονται διαφορετικοί οδηγοί για κάθε λειτουργικό σύστημα που χρησιμοποιεί ο χρήστης. Επίσης υπάρχουν οδηγοί προς διαχειριστές συστημάτων Openstack, οδηγοί για την ασφάλεια του νέφους κτλ. Τέλος, υπάρχει το forum του Openstack και το κανάλι IRC όπου ο κάθε ενδιαφερόμενος μπορεί να αναζητήσει πληροφορίες και βοήθεια.

Στα κύρια χαρακτηριστικά του Openstack παραπάνω είδαμε την διαχείριση εικόνων με την χρήση εικονικών μηχανών. Το Openstack δίνει την δυνατότητα δημιουργίας , αποθήκευσης και διαχείρισης εικόνων με την χρήση εικονικών μηχανών**(Virtual Machine Image Management)**. Το χαρακτηριστικό αυτό, αποτελεί και το κύριο μέλημα της πτυχιακής μας εργασίας και για αυτόν τον λόγο το αναφέρουμε ως ένα ξεχωριστό, θετικό χαρακτηριστικό του Openstack.

Ως επίλογος, θα λέγαμε ότι το Openstack αποτελεί μια πλήρη και ολοκληρωμένη λύση στον χώρο του υπολογιστικού νέφους με το μόνο, όχι και τόσο, αρνητικό στοιχείο την πολύ γρήγορη ανάπτυξη του που έχει ως αποτέλεσμα καινούρια χαρακτηριστικά και δυνατότητες να έρχονται στο προσκήνιο, φέρνοντας μαζί τους και τυχόν προβλήματα τα οποία, όμως, αντιμετωπίζονται.

3.1.4.1 Η ασφάλεια στο Openstack

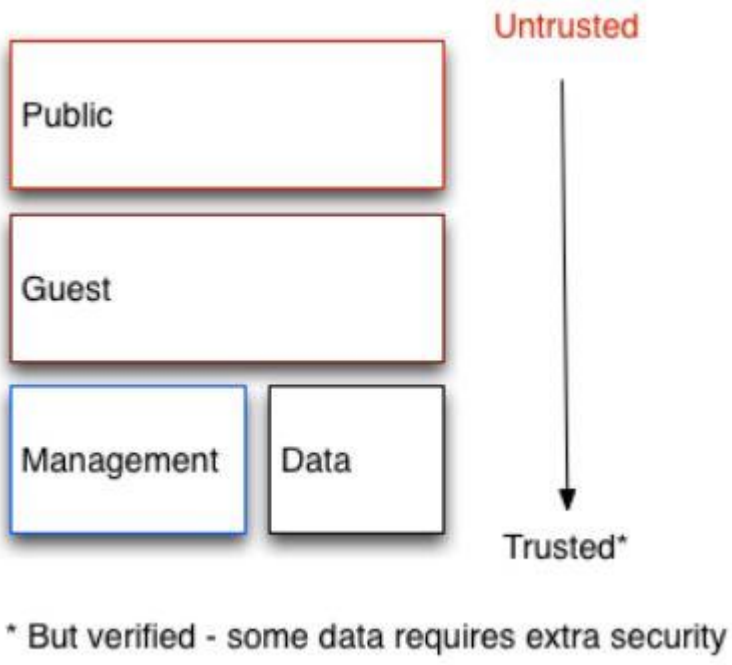
Ένας αφηρημένος ορισμός για το νέφος μπορεί να είναι μια συλλογή από λογικά συστατικά της λειτουργίας, των χρηστών και των κοινών ανησυχιών ασφαλείας, τις οποίες και αποκαλούμε **“Τομείς Ασφαλείας (Security Domains)”**. Οι φορείς των κινδύνων, ταξινομούνται ανάλογα με το κίνητρο τους και της πρόσβασης τους σε πόρους. Στόχος είναι η παροχή μιας αίσθησης αυτών των ανησυχιών ασφαλείας σε σχέση με το κάθε Domain ανάλογα με το ρίσκο και την ευπάθεια του.

Ένα Security Domain περιλαμβάνει τους χρήστες, εφαρμογές, servers ή δίκτυα που μοιράζονται κοινές απαιτήσεις και προσδοκίες εμπιστοσύνηw μέσα σε ένα σύστημα. Τυπικά, έχουν τις ίδιες απαιτήσεις αυθεντικοποίησης και εξουσιοδότησης και τους ίδιους χρήστες. Σε γενικές γραμμές υπάρχουν τέσσερα διακριτά Security Domains που σχηματίζουν το βασικά και ελάχιστο που απαιτείται για την ανάπτυξη ενός ασφαλούς Openstack νέφους. Αυτά τα Security Domains είναι:

1. Public
2. Guest
3. Management
4. Data

Η επιλογή αυτών των Domains γίνεται επειδή μπορούν να χαρτογραφούνται ανεξάρτητα ή να συνδυάζονται ώστε να αντιπροσωπεύουν την πλειοψηφία των πιθανών έμπιστων περιοχών εντός ενός ήδη αναπτυγμένου Openstack νέφους. Για παράδειγμα, κάποια τοπολογία μιας εφαρμογής Openstack μπορεί να αποτελείται από έναν συνδυασμό των Guest και Data Domains σε έναν φυσικό δίκτυο, ενώ άλλες τοπολογίες μπορεί να έχουν τα Domains αυτά σε ξεχωριστά δίκτυα. Σε κάθε περίπτωση όμως, ο χειριστής του νέφους θα πρέπει να είναι ενήμερος και προσεκτικός με την ασφάλεια του νέφους. Τα Security Domains

θα πρέπει να χαρτογραφούνται ανάλογα με την ειδική τοπολογία του κάθε Openstack νέφους. Τα domains και οι απαιτήσεις εμπιστοσύνης τους εξαρτώνται από το αν το instance του νέφους είναι public,private ή hybrid, όπως μπορούμε να δούμε και στην εικόνα που ακολουθεί.



3.8 Η εμπιστοσύνη ανάλογα με το είδος του νέφους

Το **public Security Domain** είναι μια εντελώς μη έμπιστη περιοχή της υποδομής του νέφους. Μπορεί να παραπέμψει στο Διαδίκτυο στο σύνολό του ή σε δίκτυα στα οποία δεν έχουμε καμία εξουσιοδότηση. Όλα τα δεδομένα που διακινούνται εντός του τομέα αυτού με απαιτήσεις εμπιστευτικότητας και ακεραιότητας, θα πρέπει να προστατεύονται με τα κατάλληλα εργαλεία. Αυτός ο τομέας θα πρέπει να θεωρείται αναξιόπιστος.

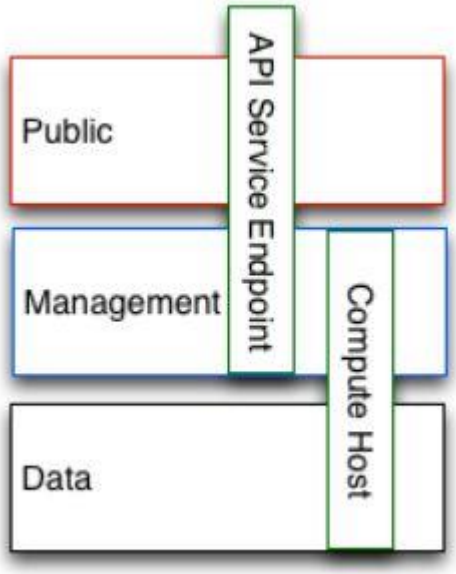
Το **guest Security Domain**, συνήθως χρησιμοποιείται για την υπολογιστική κυκλοφορία μεταξύ των instances. Το domain αυτό χειρίζεται τα δεδομένα που δημιουργούνται από τα instances του νέφους και όχι τα δεδομένα των υπηρεσιών που υποστηρίζουν την λειτουργία του νέφους (π.χ. API calls). Πάροχοι public και Private νεφών που δεν έχουν αυστηρούς ελέγχους στην χρήση των instances ή επιτρέπουν την απεριόριστη πρόσβαση των εικονικών μηχανών στο Διαδίκτυο, θα πρέπει να θεωρούν το domain αυτό ως μη αξιόπιστο. Οι πάροχοι των private νεφών μπορεί να θεωρήσουν αυτό το domain ως εσωτερικό και έμπιστό μόνο με

την χρήση των κατάλληλων εργαλείων και ελέγχων έτσι ώστε τα instances να είναι ασφαλή και έμπιστα.

Το **Management Security domain** είναι ο τόπος που οι υπηρεσίες αλληλεπιδρούν. Μερικές φορές αναφέρεται και ως το επίπεδο ελέγχου. Τα δίκτυα σε αυτόν τον τομέα μεταφέρουν εμπιστευτικά δεδομένα όπως για παράδειγμα, ονόματα και κωδικοί χρηστών. Η κυκλοφορία διοίκησης και ελέγχου του νέφους, συνήθως, διακινείται εντός αυτού του τομέα, ο οποίος απαιτεί ισχυρές δικλείδες ακεραιότητας. Η πρόσβαση σε αυτόν τον τομέα θα πρέπει να είναι εξαιρετικά περιορισμένη και να παρακολουθείται. Στις περισσότερες περιπτώσεις, αυτό το domain θεωρείται αξιόπιστο. Όμως σε μερικές περιπτώσεις Openstack νεφών, το domain αυτό γεφυρώνεται με άλλα domains και ενδεχομένως έτσι μειώνουν το επίπεδο εμπιστευτικότητας που μπορεί να έχει κάποιος για αυτό.

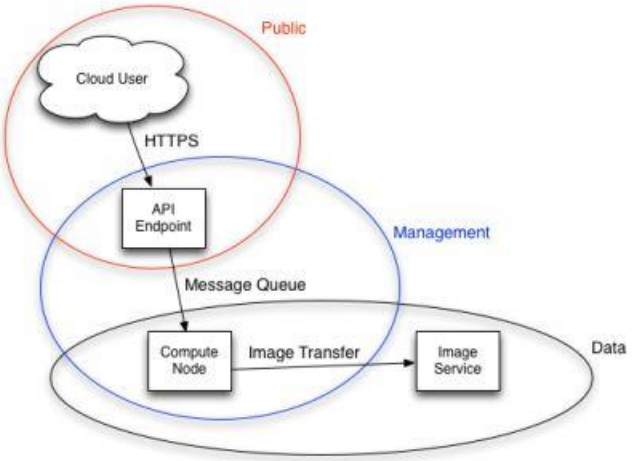
Το **Data Security Domain** συνδέεται κυρίως με τις πληροφορίες που αφορούν τις υπηρεσίες αποθήκευσης δεδομένων εντός του Openstack. Το μεγαλύτερο μέρος των δεδομένων που διακινούνται εντός αυτού του τομέα απαιτούν υψηλά επίπεδα ακεραιότητας και εμπιστευτικότητας. Σε μερικές περιπτώσεις, ανάλογα με τον τύπο του νέφους που έχει αναπτυχθεί μπορεί να υπάρξουν και υψηλές απαιτήσεις της διαθεσιμότητας του. Το επίπεδο αξιοπιστίας του δικτύου αυτού σε μεγάλο βαθμό εξαρτάται στις αποφάσεις που θα παρθούν για την ανάπτυξη του και ως εκ τούτου δεν υπάρχει προεπιλεγμένο επίπεδο εμπιστοσύνης.

Όσον αφορά τα domains,υπάρχει ένα συστατικό που υπάρχει στο εσωτερικό των domains και τα συνδέει μεταξύ τους. Το συστατικό αυτό λέγεται **Γέφυρα (Bridge)**. Κάθε γέφυρα που ενώνει domain με διαφορετικά επίπεδα αξιοπιστίας θα πρέπει να ρυθμίζεται πολύ προσεκτικά. Οι γέφυρες είναι συχνά τα αδύνατα σημεία στην αρχιτεκτονική ενός δικτύου. Μια γέφυρα πρέπει πάντοτε να ρυθμίζεται ώστε να πληρεί τις απαιτήσεις ασφαλείας του υψηλότερου επιπέδου εμπιστοσύνης ανεξαρτήτως των τομέων που γεφυρώνει. Σε πολλές περιπτώσεις, οι έλεγχοι ασφαλείας των γεφυρών πρέπει να είναι το πρωταρχικό μέλημα, λόγω της πιθανότητας επίθεσης.



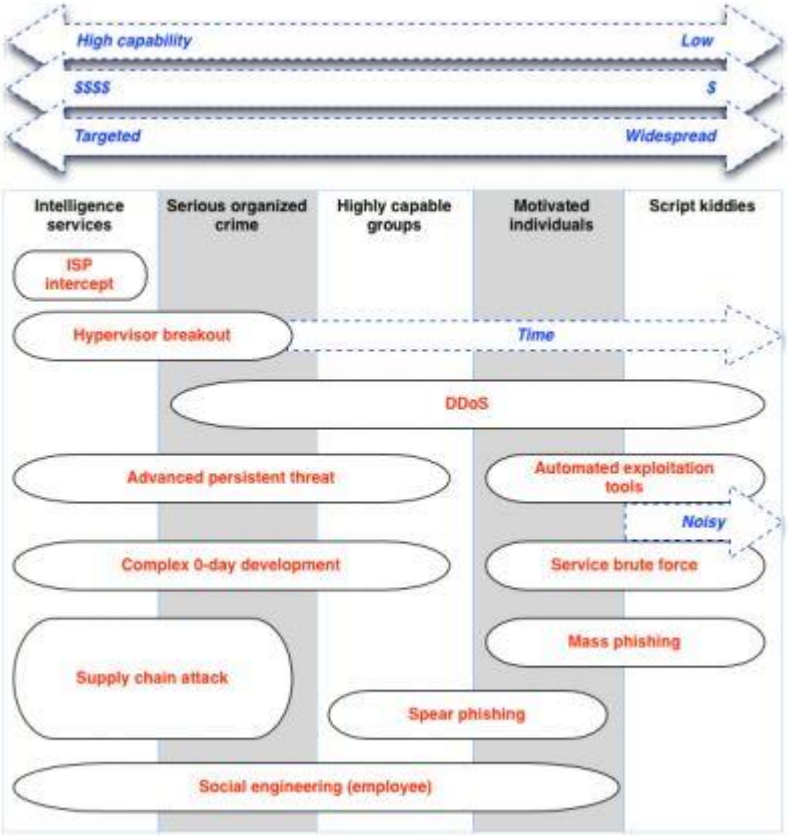
3.9 Γεφύρωση Domains

Η εικόνα παραπάνω δείχνει έναν υπολογιστικό κόμβο ο οποίος γεφυρώνει το data και το management domain, ως εκ τούτου ο κόμβος θα πρέπει να ρυθμιστεί ώστε να ανταποκρίνεται στις απαιτήσεις ασφαλείας του management domain. Ομοίως, το API Endpoint του σχήματος γεφυρώνει το αναξιόπιστο public domain με το management domain. Έτσι το management domain θα πρέπει να ρυθμιστεί ώστε να παρέχει προστασία απέναντι σε επιθέσεις που μπορεί να προέρχονται από το public.



3.10 API Endpoint Bridging

Σε μερικές περιπτώσεις, οι υπεύθυνοι της ανάπτυξης του νέφους είναι πιθανόν να πρέπει να εξετάσουν το ενδεχόμενο της διασφάλισης μιας γέφυρας με ένα υψηλότερο πρότυπο από αυτά που υπάρχουν ήδη στα domains. Στο παραπάνω παράδειγμα της εικόνας υπάρχει ένα API Endpoint. Δυνητικά κάποιος θα μπορούσε να στοχεύσει το παρόν API Endpoint προερχόμενος από το public domain με σκοπό να θέσει σε κίνδυνο το νέφος ή να αποκτήσει πρόσβαση στο management domain. Ο σχεδιασμός του Openstack είναι τέτοιος ώστε ο διαχωρισμός των domains είναι δύσκολος, καθώς οι βασικές υπηρεσίες του συνήθως γεφυρώνουν το λιγότερο 2 domains. Έτσι θα πρέπει να δοθεί ιδιαίτερη προσοχή όταν εφαρμόζουμε ελέγχους ασφαλείας στα domains μας.

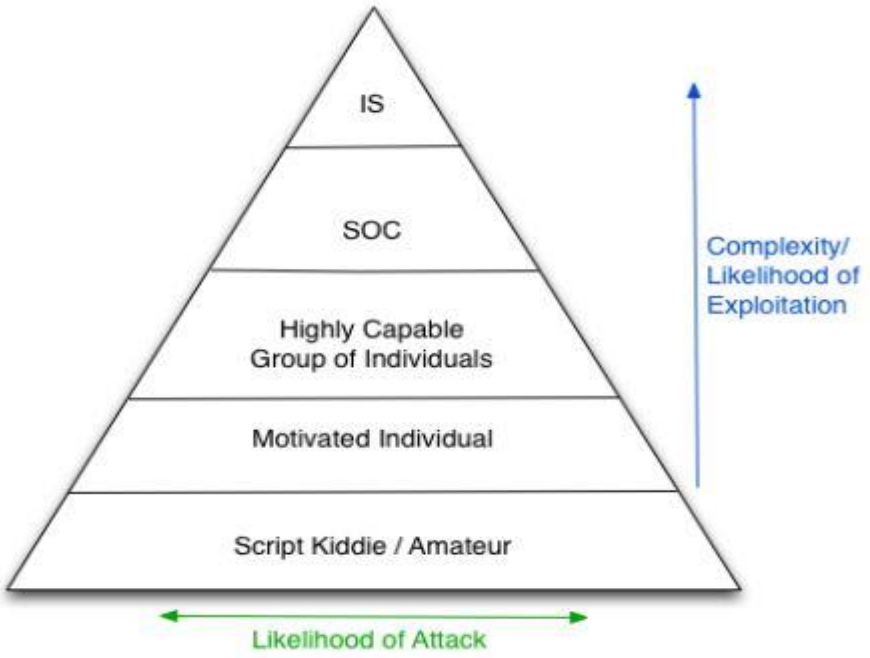


3.11 Τύποι Επιθέσεων

Στην παραπάνω εικόνα βλέπουμε τα είδη των επιθέσεων που μπορεί να γίνουν σε ένα νέφος. Βέβαια, μπορεί να υπάρχουν και εξαιρέσεις σε αυτό το διάγραμμα αλλά γενικώς, περιγράφει τα είδη επίθεσης ανάλογα με την κατηγορία στην οποία ανήκουν. Οι κατηγορίες αυτές είναι πέντε :

1. Intelligence Services
2. Serious Organized Crime
3. Highly Capable Groups
4. Motivated Individuals
5. Script Kiddies

Επίσης το παραπάνω διάγραμμα μπορεί να βοηθήσει ώστε να αποφασίσουμε ενάντια σε ποιες επιθέσεις θα πρέπει να λάβουμε μέτρα προστασίας. Για δημόσια, εμπορικά νέφη θα μπορούσαμε να περιλάβουμε μέτρα πρόληψης απέναντι στην κατηγορία Serious Organized Crime. Επίσης, για όσους αναπτύσσουν ιδιωτικά νέφη για κυβερνητική χρήση, θα πρέπει να χρησιμοποιήσουν αυστηρότερους προστατευτικούς μηχανισμούς, συμπεριλαμβανομένων προστατευόμενες εγκαταστάσεις και αλυσίδες εφοδιασμού. Στην εικόνα 3.11 που ακολουθεί μπορούμε να δούμε πως σχετίζεται η πολυπλοκότητα της έκθεσης του συστήματος και η πιθανότητα της επίθεσης ανάλογα με την κατηγορία στην οποία ανήκει η εκάστοτε επίθεση.



3.12 Κατηγορίες Επιθέσεων και πιθανότητα Επίθεσης και Πολυπλοκότητας

Το Openstack όπως είναι λογικό παρέχει και ασφαλή επικοινωνία εντός του νέφους. Υπάρχουν απαιτήσεις ασφάλειας για την διασφάλιση του απορρήτου και

της ακεραιότητας στην κίνηση δεδομένων σε ένα δίκτυο ενός Openstack νέφους. Γενικά, αυτό επιτυγχάνεται με την χρήση κρυπτογραφικών μέτρων, όπως το TLS πρωτόκολλο. Σε μία τυπική και απλή υλοποίηση ενός νέφους με την χρήση του Openstack, όλη η κίνηση που μεταφέρεται μέσω δημοσίων δικτύων είναι ασφαλής. Βέβαια, η καλύτερη πρακτική ασφαλείας υπαγορεύει ότι και η εσωτερική κίνηση θα πρέπει να είναι ασφαλείς. Δεν αρκεί να βασιζόμαστε στον διαχωρισμό των Security Domains για την προστασία. Στην περίπτωση που κάποιος επιτεθεί, αποκτήσει πρόσβαση στον hypervisor ή σε πόρους του host, θέτει σε κίνδυνο ένα API Endpoint, ή οποιαδήποτε άλλη υπηρεσία, δεν θα πρέπει να είναι εύκολο να αλλοιώσει ή να υποκλέψει μηνύματα και εντολές γιατί σε αυτήν την περίπτωση θα θέσει σε κίνδυνο την ολική διαχείριση του νέφους. Όλα τα domains θα πρέπει να ασφαλίζονται με TLS, συμπεριλαμβανομένων και των υπηρεσιών του management domain, καθώς και η επικοινωνία των υπηρεσιών στο εσωτερικό του δικτύου. Το TLS παρέχει μηχανισμούς για την εξασφάλιση του έλεγχου ταυτότητας, της εμπιστευτικότητας και της ακεραιότητας της επικοινωνίας των χρηστών με τις υπηρεσίες του Openstack, καθώς και των υπηρεσιών μεταξύ τους.

Παρακάτω ακολουθούν κάποια παραδείγματα ρυθμίσεων που συνιστώνται για την πραγματοποίηση TLS με κάποιους από τους πιο δημοφιλείς web servers και TLS terminators.

```
ciphers = "HIGH:!RC4:!MD5:!aNULL:!eNULL:!EXP:!LOW:!MEDIUM"
```

ή

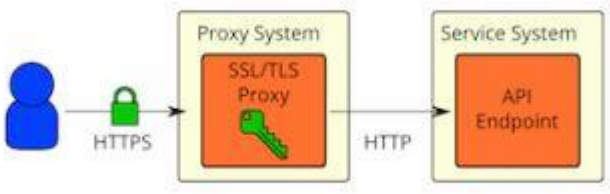
```
ciphers="kEECDH:kEDH:kRSA:HIGH:!RC4:!MD5:!aNULL:!eNULL:!EXP:! LOW:!MEDIUM"
```

Οι επιλογές κρυπτογράφησης του String χωρίζονται με “:”, ενώ το “!” παρέχει άρνηση του αμέσως επόμενου στοιχείου. Η ταξινόμηση των στοιχείων δείχνει την προτίμηση, εκτός και αν παρακαμφθούν με ποσοδείκτες όπως το HIGH. Ας ρίξουμε μια ματιά στα στοιχεία των παραπάνω παραδειγμάτων.

HIGH	Επιλογή της υψηλότερης δυνατής ασφάλειας κρυπτογράφησης στην φάση της διαπραγμάτευσης. Συνήθως με κλειδιά μήκους 128 bits ή περισσότερα.
!RC4	Δεν χρησιμοποιείται το RC4.
!MD5	Δεν χρησιμοποιείται το MD5 καθώς δεν είναι ανθεκτικό στις συγκρούσεις και δεν είναι αποδεκτό για Message Authentication Codes (MAC) ή υπογραφές.
!aNULL:!eNULL	Δεν επιτρέπει την χρήση καθαρού κειμένου.
!EXP	Αποτρέπει την εξαγωγή αλγορίθμων κρυπτογράφησης, η οποία λόγω σχεδιασμού τείνει να είναι αδύναμη χρησιμοποιώντας κλειδιά συνήθως 40 και 56 bits.
!LOW:!MEDIUM	Δεν επιτρέπει χαμηλούς(56 ή 64 bits μήκος κλειδιών) ή μέσους(128 bits μήκος κλειδιών) αλγόριθμους κρυπτογράφησης, λόγω της ευπάθειας τους σε brute force επιθέσεις (παράδειγμα ο 2-DES).
Protocols	Τα πρωτόκολλα ενεργοποιούνται ή απενεργοποιούνται μέσω της εντολής SSL_CTX_set_options.

Είναι γενικά αποδεκτό ότι είναι καλύτερο να κρυπτογραφούμε ευαίσθητα δεδομένα όσο το δυνατόν νωρίτερα και να τα αποκρυπτογραφούμε όσο το δυνατόν αργότερα. Παρά την βέλτιστη αυτή πρακτική, φαίνεται πως είναι διαδεδομένο να χρησιμοποιούμε ένα SSL/TLS proxy πριν από τις υπηρεσίες

Openstack και να χρησιμοποιούμε καθαρή επικοινωνία αργότερα όπως φαίνεται στην εικόνα που ακολουθεί.



3.13 SSL/TLS proxy

Από την παραπάνω εικόνα δημιουργούνται κάποιες ανησυχίες που αφορούν την χρήση του proxy.

- Το Native SSL/TLS στις υπηρεσίες του Openstack δεν λειτουργεί και δεν κλιμακώνεται τόσο καλά όσο άλλα SSL proxies (ιδιαίτερα εφαρμογές Python όπως το Eventlet).
- Το Native SSL/TLS στις υπηρεσίες του Openstack δεν ελέγχει και δεν ελέγχεται τόσο καλά όσο άλλες πιο δοκιμασμένες λύσεις.
- Το Native SSL/TLS είναι δύσκολο στην ρύθμιση (Δεν υπάρχει καλή βιβλιογραφία, δεν είναι δοκιμασμένο και συνεπές με όλες τις υπηρεσίες)
- Διαχωρισμός προνομίων (Οι διεργασίες των υπηρεσιών του Openstack δεν πρέπει να έχουν άμεση πρόσβαση σε ιδιωτικά κλειδιά που χρησιμοποιούνται για SSL/TLS)
- Χρειάζεται έλεγχους της κυκλοφορίας των δεδομένων για την εξισορρόπηση του φόρτου.

Όλα τα παραπάνω είναι βάσιμες ανησυχίες, αλλά καμία από αυτές δεν εμποδίζει την χρήση SSL/TLS στο δίκτυο της διαχείρισης.

Το Openstack παρέχει επίσης και ασφάλεια σε θέματα αυθεντικοποίησης. Η υπηρεσία που είναι υπεύθυνη για την ταυτοποίηση παρέχει πολλές μεθόδους ώστε να αυτή να επιτευχθεί. Μπορεί να γίνει με την χρήση ονόματος και κωδικού χρήστη, LDAP καθώς και με μεθόδους εξωτερικής ταυτοποίησης. Μετά την επιτυχή ταυτοποίηση, η υπηρεσία παρέχει στον χρήστη ένα token που χρησιμοποιείται για τα μεταγενέστερα αιτήματα του . Το TLS παρέχει τον έλεγχο

ταυτότητας μεταξύ υπηρεσιών και χρηστών χρησιμοποιώντας X.509 πιστοποιητικά.

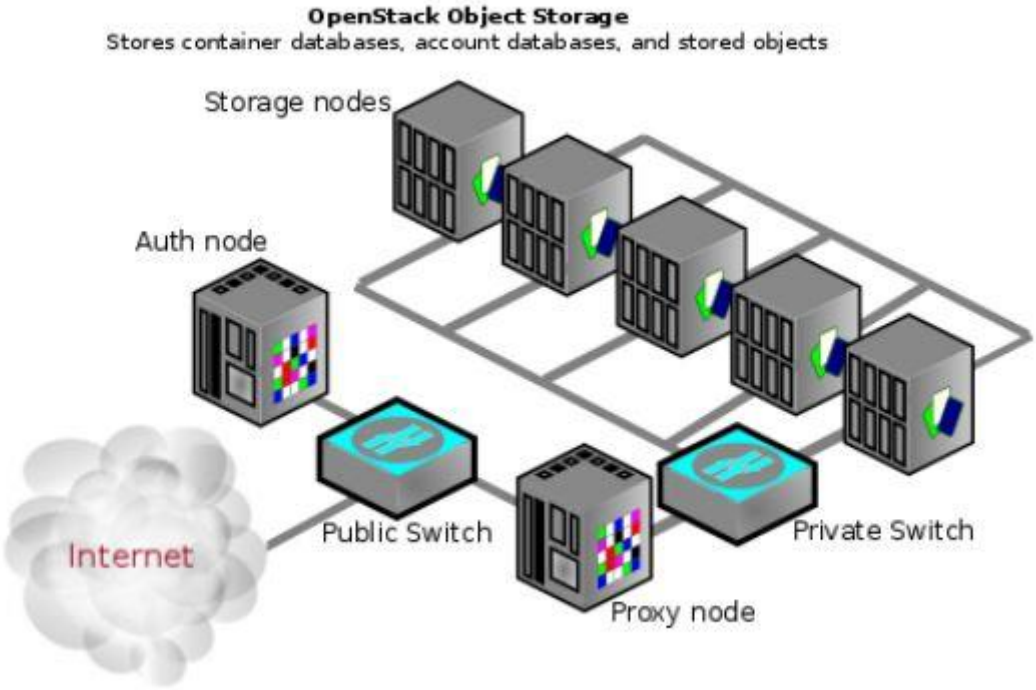
Η υπηρεσία ταυτοποίησης δεν παρέχει κάποια μέθοδο για να περιορίζει την πρόσβαση σε λογαριασμούς χρηστών μετά από έναν αριθμό ανεπιτυχών προσπαθειών για σύνδεση. Αυτές οι απόπειρες πιθανότατα μπορεί να είναι και επιθέσεις brute force. Για την αποφυγή αυτού, λύση αποτελεί η χρήση ενός εξωτερικού συστήματος ταυτοποίησης που θα αποκλείει λογαριασμούς που προσπαθούν να συνδεθούν ανεπιτυχώς. Αν τώρα δεν υπάρχει η επιλογή της πρόληψης, θα πρέπει να υπάρχει ανίχνευση για να μετριάσει την βλάβη. Η ανίχνευση περιλαμβάνει την συχνή επιθεώρηση των logs ελέγχου πρόσβασης για τον εντοπισμό με εξουσιοδοτημένων προσπαθειών για πρόσβαση σε λογαριασμούς χρηστών. Επίσης πιθανή αποκατάσταση μια τέτοιας περίπτωσης θα μπορούσε να περιλαμβάνει την ανασκόπηση της δυναμικής των κωδικών πρόσβασης και τον δικτυακό αποκλεισμό της πηγής της επίθεσης με την χρήση κανόνων στο τοίχος προστασίας.

Συνεχίζοντας, το Openstack έχει προνοήσει και για την ασφάλεια της Web διεπαφής που ονομάζεται Dashboard. Το Dashboard πρέπει να αναπτύσσεται ως μια **Web Service Gateway Interface (WSGI)** εφαρμογή πίσω από έναν HTTPS proxy όπως ο Apache ή ο ginx. Το Dashboard είναι καλό να αναπτυχθεί με την βοήθεια ενός ασφαλούς HTTPS server που θα χρησιμοποιεί ένα έγκυρο πιστοποιητικό αναγνωρισμένο από την Αρχή Πιστοποίησης (CA). Το Dashboard βασίζεται σε ένα κοινό SECRET_KEY για κάποιες λειτουργίες ασφαλείας. Είναι ένα τυχαίο String που δημιουργείται, με μήκος 64 χαρακτήρων και διαμοιράζεται σε όλα τα ενεργά Dashboard instances. Επίσης τα Session Cookies θα πρέπει να ρυθμιστούν σε κατάσταση HTTPONLY. Με την ρύθμιση να έχει την παρακάτω μορφή.

SESSION_COOKIE_HTTPONLY = True

Σε αντίθεση με παρόμοια συστήματα, το Openstack Dashboard επιτρέπει την χρήση όλων των Unicode χαρακτήρων. Αυτό σημαίνει πως οι developers έχουν μικρότερο περιθώριο λάθους ώστε να δώσουν δικαίωμα για **Cross Site Scripting(XSS)** επιθέσεις.

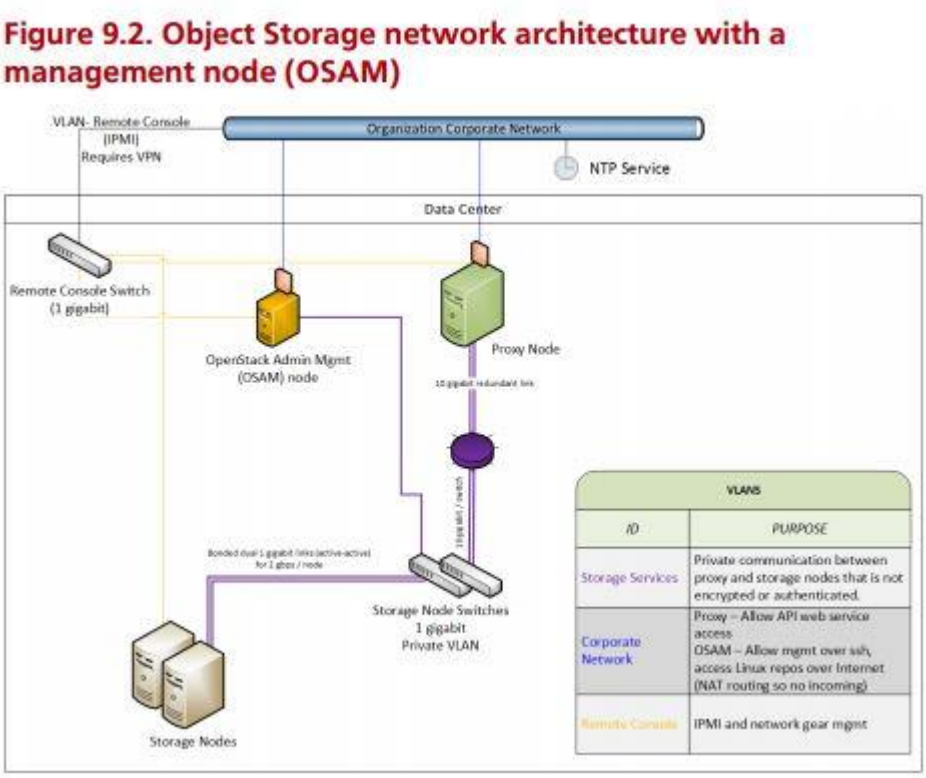
Συνεχίζοντας με την ασφάλεια που παρέχει το Openstack, μια άλλη πτυχή αυτής είναι η ασφάλεια των αντικειμένων αποθήκευσης (Object Storage) και η υπηρεσία που είναι υπεύθυνη για αυτόν τον σκοπό(Swift). Στην εικόνα που ακολουθεί μπορούμε να δούμε ένα παράδειγμα με την εφαρμογή του Swift.



3.14 Παράδειγμα Υλοποίησης Swift

Η εγκατάσταση του Object Storage δεν είναι αναγκαστικό να γίνεται στο Internet, μιας και μπορεί να γίνει σε ένα ιδιωτικό νέφος με το “Δημόσιο Switch” να είναι μέρος της υλοποίησης του εσωτερικού δικτύου του οργανισμού.

Το πρώτο πράγμα για την ασφάλεια του Object Storage είναι η ασφάλεια του δικτύου. Το πρωτόκολλο rsync χρησιμοποιείται ανάμεσα σε κόμβους του Object Storage για την αναπαραγωγή των δεδομένων με σκοπό την υψηλή διαθεσιμότητα τους. Επίσης, η υπηρεσία proxy επικοινωνεί με την υπηρεσία αποθήκευσης όταν δεδομένα διακινούνται μεταξύ του νέφους και του χρήστη. Στην Εικόνα 3.13 βλέπουμε ένα Private Switch. Ο λόγος ύπαρξης του είναι διότι το Object Storage δεν εφαρμόζει κρυπτογράφηση και ταυτοποίηση στην εσωτερική επικοινωνία των κόμβων του. Έτσι αυτό το data domain πρέπει να είναι διαχωρισμένο από τα υπόλοιπα δίκτυα Openstack.



3.15 Αρχιτεκτονική Αποθήκευσης Αντικειμένων με κόμβο Διαχείρισης – Object Storage Architecture Management (OSAM)

Υπάρχουν συγκεκριμένες υπηρεσίες αποθήκευσης που εμπεριέχονται στο Object Storage. Αυτές οι υπηρεσίες καθώς και οι θύρες στις οποίες δέχονται και στέλνουν δεδομένα παραθέτονται στον ακόλουθο πίνακα.

Πίνακας 3.1 Υπηρεσίες αποθήκευσης

Service name	Port	Type
Account service	6002	TCP
Container service	6001	TCP
Object service	6000	TCP
Rsync[a]	873	TCP

[a]If ssync is used instead of rsync, the Object service port is used for maintaining durability.

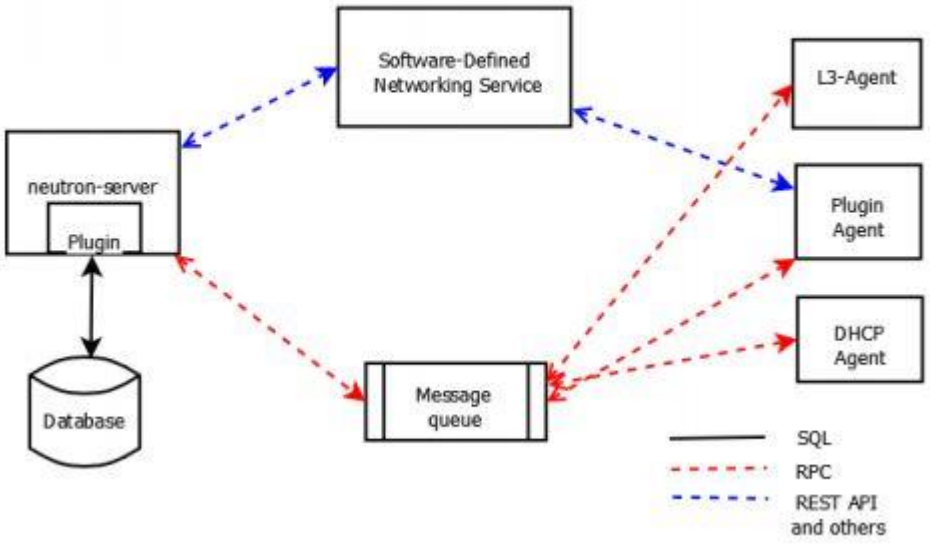
Όπως προείπαμε ταυτοποίηση δεν λαμβάνει χώρα στους κόμβους αποθήκευσης. Εάν κάποιος ήταν σε θέση να συνδεθεί σε έναν τέτοιο κόμβο ή σε μία από τις θύρες του πίνακα παραπάνω τότε θα μπορούσε να έχει πρόσβαση ή να αλλάξει τα δεδομένα χωρίς ταυτοποίηση. Για την διασφάλιση ενός τέτοιου γεγονότος θα πρέπει όπως αναφέραμε να χρησιμοποιήσουμε ένα ιδιωτικό δίκτυο αποθήκευσης.

Στην συνέχεια έχουμε την ασφάλεια της υπηρεσίας δικτύωσης του Openstack περιβάλλοντος. Η υπηρεσία αυτή είναι αυτόνομη. Συχνά αναπτύσσει διάφορες διεργασίες μεταξύ των κόμβων. Αυτές οι διεργασίες όχι μόνο αλληλεπιδρούν μεταξύ τους, αλλά και με άλλες υπηρεσίες. Κύρια επεξεργασία του Openstack Networking είναι το **neutron-server**, ένα Python daemon που χρησιμοποιεί το API της δικτύωσης για να προωθεί αιτήσεις σε άλλες υπηρεσίες για επεξεργασία.

Τα συστατικά που αποτελούν το Openstack Networking είναι:

1. **Neutron server (neutron-server and neutron-*-plugin)**. Αυτή η υπηρεσία λειτουργεί στον δικτυακό κόμβο για να εξυπηρετεί το δικτυακό API. Χρειάζεται πρόσβαση σε μια Βάση Δεδομένων για συνεχή αποθήκευση και πρόσβαση σε ένα Message Queue για ενδοεπικοινωνία.
2. **Plugin agent (neutron-*- agent)**. Λειτουργεί σε κάθε υπολογιστικό κόμβο για την διαχείριση των τοπικών εικονικών switch. Επίσης απαιτεί την χρήση Message Queuing για να λειτουργήσει σωστά.
3. **L3 agent (neutron-l3- agent)**. Παρέχει L3/NAT Προώθηση για εξωτερική δικτυακή πρόσβαση των εικονικών μηχανών. Απαιτεί την χρήση Message Queuing για να λειτουργήσει σωστά.
4. **DHCP agent (neutron-dhcpagent)**. Παρέχει λειτουργίες DHCP. Είναι ίδιο για όλα τα plugins και είναι υπεύθυνο για τις DHCP ρυθμίσεις. Απαιτεί την χρήση Message Queuing για να λειτουργήσει σωστά.
5. **Network provider services (SDN server/services)**. Παρέχει περαιτέρω δικτυακές λειτουργίες. Οι SDN υπηρεσίες μπορεί να αλληλεπιδρούν με τα neutron-server, neutron-plugin μέσω του API ή άλλων καναλιών.

Το παρακάτω σχήμα δείχνει ένα αρχιτεκτονικό διάγραμμα, καθώς και την δικτυακή ροή των συστατικών που παρατέθηκαν.



3.16 Αρχιτεκτονική και Ροή του Openstack Networking

Τέλος, πολλοί αλγόριθμοι κρυπτογράφησης είναι διαθέσιμοι για την ταυτοποίηση, εξουσιοδότηση, μεταφορά δεδομένων και προστασία του Openstack. Ακολουθεί ένας πίνακας με τα χαρακτηριστικά του κάθε αλγόριθμου καθώς και με την χρηστικότητά του.

Πίνακας 3.2 Αλγόριθμοι κρυπτογράφησης και η χρηστικότητά τους

Algorithm	Key Length	Intended Purpose	Security Function	Implementation Standard
AES	128,192,256 bits	Κρυπτογράφηση, Αποκρυπτογράφηση	Προστατευόμενη Μεταφορά Δεδομένων	RFC 4253
TDES	168 bits	Κρυπτογράφηση, Αποκρυπτογράφηση	Προστατευόμενη Μεταφορά Δεδομένων	RFC 4253
RSA	1024,2048,3072 bits	Ταυτοποίηση, Ανταλλαγή Κλειδιών	Αναγνώριση και Ταυτοποίηση, Προστατευόμενη Μεταφορά Δεδομένων	U.S. NIST FIPS PUB 186-3
DSA	L=1024, N=160 bits	Ταυτοποίηση, Ανταλλαγή Κλειδιών	Αναγνώριση και Ταυτοποίηση, Προστατευόμενη Μεταφορά Δεδομένων	U.S. NIST FIPS PUB 186-3
Serpent	128, 192, or 256 bits	Κρυπτογράφηση, Αποκρυπτογράφηση	Προστατευόμενη Μεταφορά Δεδομένων	www.cl.cam.ac.uk/ ~rja14/Papers/ serpent.pdf
Twofish	128, 192, or 256 bit	Κρυπτογράφηση, Αποκρυπτογράφηση	Προστατευόμενη Μεταφορά Δεδομένων	www.schneier.com/ paper-twofishpaper.htm
SHA-1	-	Message Digest	Προστατευόμενη Μεταφορά Δεδομένων, Αναγνώριση και	U.S. NIST FIPS PUB 180-3

			Ταυτοποίηση	
SHA-2 (224, 256, 384, or 512 bits)	-	Message Digest	Προστατευόμενη Μεταφορά Δεδομένων, Αναγνώριση και Ταυτοποίηση	U.S. NIST FIPS PUB 180-3

Αφού είδαμε και μελετήσαμε τα τέσσερα αυτά εργαλεία που μας δίνουν την δυνατότητα ανάπτυξης του δικού μας υπολογιστικού νέφους, στο αμέσως υποκεφάλαιο θα τα συγκρίνουμε. Παραθέτουμε έναν συγκριτικό πίνακα με χαρακτηριστικά ενός νέφους καθώς την υποστήριξη ή μη από την κάθε πλατφόρμα ξεχωριστά. Επίσης αμέσως μετά τον πίνακα ακολουθεί μια μικρή επεξήγηση για το κάθε χαρακτηριστικό που παρουσιάζεται στον παρακάτω πίνακα, κάποιες αριθμημένες σημειώσεις σε σχέση με τα χαρακτηριστικά και την εκάστοτε πλατφόρμα και φυσικά η σύγκριση των επιλογών μας ανάλογα το εκάστοτε χαρακτηριστικό.

3.2 Η Σύγκριση των Επιλογών της

Πίνακας 3.3 Σύγκριση χαρακτηριστικών

Χαρακτηριστικά	CloudStack	Eucalyptus	Vcloud Director	Openstack
AD Integration	+/-(1)	-	+	+
Management Console	+	-(2)	+	+
Web Access to the VM Console	+	-(3)	+	+
API	+	+	+	+
Multi-Role Support	+	+	+	+
VLANS	+	+	+	+
Hypervisor Agnostic	+	+	+	+
Easy Template Creation Process	+	-	+	+
Snapshots	+	+	+	+
Resource Over Provisioning and Limits	+	+	+	+
Alerts and Notifications	+	-	+	+
Volumes	+	+	+	+
Guest OS Preferences(4)	+	-	+	+
Host Maintenance with Live Migration	+	-	+	-
Free	+	+/-(5)	-	+
High Availability Cloud	+	+	+	+

Component				
Implementation Complexity[6]	+	-	+	-

Το 1ο χαρακτηριστικό είναι το **AD Integration**. Αποτελεί την εφαρμογή και την υποστήριξη Active Directory. Περιεκτικά το Active Directory μπορεί να οριστεί ως μια υπηρεσίας καταλόγου. Ρόλος του AD είναι η αυθεντικοποίηση και η κατάλληλη εξουσιοδότηση των χρηστών εφαρμόζοντας της κατάλληλες πολιτικές ασφαλείας που έχουν οριστεί.

Ως **Management Console** μπορούμε να ορίσουμε την χρησιμοποίηση της κονσόλας, της εκάστοτε πλατφόρμας, σε ένα οποιοδήποτε τερματικό, η οποία της δίνει την δυνατότητα διαχείρισης, απομακρυσμένης ή μη, του υπολογιστικού της νέφους.

Το **Web Access to the VM Console** αποτελεί την δυνατότητα πρόσβασης σε οποιαδήποτε εικονική μηχανή έχουμε δημιουργήσει, μέσω της απλού φυλλομετρητή. Αυτό γίνεται πληκτρολογώντας συνήθως την IP διεύθυνση της εικονικής μηχανής στην γραμμή διευθύνσεων του φυλλομετρητή. Απαραίτητη προϋπόθεση είναι η δυνατότητα αυτή να υποστηρίζεται από τον φυλλομετρητή που χρησιμοποιούμε.

Το **API (Application Programming Interface)** ως χαρακτηριστικό ορίζεται ως η δυνατότητα της κάθε πλατφόρμας να της παρέχει το δικό της. Με της υπηρεσίες και της λειτουργίες που της προσφέρει το API κάθε πλατφόρμας μπορούμε να αναπτύξουμε και τα δικά της συστήματα βασισμένα σε αυτό.

Συνεχίζοντας, ως **Multi-Role Support** μπορεί να οριστεί η δημιουργία πολλαπλών ρόλων για της χρήστες. Κάθε διαφορετικός ρόλος μπορεί να έχει διαφορετικά δικαιώματα και δυνατότητες εντός του συστήματος. Ένα πολύ απλό παράδειγμα διαφορετικών ρόλων είναι της μεταξύ διαχειριστή και απλού χρήστη. Με τον διαχειριστή, για παράδειγμα, να μπορεί να δημιουργεί και να διαγράφει εικόνες λειτουργικών συστημάτων από το περιβάλλον χρήσης, ενώ ο απλός χρήστης να έχει μόνο το δικαίωμα να χρησιμοποιεί της ήδη υπάρχουσες εικόνες.

Hypervisor Agnostic είναι το χαρακτηριστικό της κάθε πλατφόρμας να υποστηρίζει οποιονδήποτε hypervisor και να είναι λειτουργική ανεξαρτήτως του που hypervisor θα χρησιμοποιηθεί.

Easy Template Creation Process, ορίζεται ως η ευκολία στην δημιουργία Templates. Το Template στα συστήματα νέφους, που πάνω της τρέχουν εικονικά μηχανήματα ορίζεται η διαδικασία όπου:

1. Δημιουργούμε μια εικονική μηχανή με το Λειτουργικό Σύστημα που επιθυμούμε, κάνοντας της επιθυμητές ρυθμίσεις.
2. Σταματάμε την λειτουργία της μηχανής.
3. Τέλος, μετατρέπουμε την μονάδα δίσκου της μηχανής σε Template.

Της μπορούμε να δημιουργήσουμε templates από snapshots μηχανών που ήδη έχουμε δημιουργήσει.

Έκτο χαρακτηριστικό βάσει του πίνακα παραπάνω είναι αν η πλατφόρμα υποστηρίζει **VLANs**. Πολύ σύντομα θα αναφέρουμε ότι η χρήση εικονικών τοπικών δικτύων γίνεται για την διαχείριση του μεγέθους των broadcast domain με χρήση switch.

Αμέσως επόμενο χαρακτηριστικό είναι τα **Snapshots**. Όταν δημιουργούμε ένα εικονικό μηχάνημα με το επιθυμητό λειτουργικό σύστημα, ξεκινάμε με την εγκατάσταση του, της θα κάναμε με τον προσωπικό υπολογιστή της. Στην συνέχεια, αφού το εικονικό μηχάνημα είναι έτοιμο της χρήση μπορούμε να εγκαταστήσουμε τα προγράμματα που θέλουμε. Αν πάρουμε Snapshot της εικονικής μηχανής θα μπορούμε από εκεί και πέρα να έχουμε διαθέσιμα το λειτουργικό σύστημα και τα εγκατεστημένα προγράμματα της ήταν όταν δημιουργήσαμε το Snapshot.

Παρακάτω έχουμε το **Resource Over Provisioning and Limits**. Το χαρακτηριστικό αυτό έχει να κάνει με την διαχείριση των διαθέσιμων πόρων (CPU και RAM) του συστήματος, έτσι ώστε να μπορούμε να υπολογίζουμε πόσα εικονικά μηχανήματα μπορούν να λειτουργήσουν σε κάθε cluster. Το χαρακτηριστικό αυτό μπορεί να αναπαρασταθεί από έναν αριθμό που προκύπτει από μία απλή αναλογία που ορίζεται από τον διαχειριστή του συστήματος. Για παράδειγμα στο CloudStack η αναλογία 1 είναι η αρχική και αυτό σημαίνει ότι το Over Provisioning δεν λειτουργεί.

Τα **Alerts και Notifications** αποτελούν οι διάφορες ειδοποιήσεις και τα μηνύματα που προβάλλει το σύστημα της τον χρήστη. Τα μηνύματα αυτά αφορούν οποιαδήποτε αλλαγή συμβαίνει στο σύστημα και έχει να κάνει με εικονικά μηχανήματα, κάρτες δικτύου, δίκτυα, δημόσιες IP, snapshots, templates κτλ.

Αμέσως μετά στον πίνακα υπάρχει το χαρακτηριστικό **Volumes**. Σαν Volumes μπορούμε να ορίσουμε ως ένα μέρος της αποθηκευτικής ισχύος του μηχανήματος που φιλοξενεί την πλατφόρμα. Τα Volumes αυτά χρησιμοποιούνται σαν ξεχωριστοί σκληροί δίσκοι και αναπαριστούν την αποθηκευτική ισχύ του εκάστοτε εικονικού μηχανήματος στο οποίο επισυνάπτονται. Έτσι το κάθε μηχάνημα έχει τον ιδιωτικό του αποθηκευτικό χώρο για οποιαδήποτε χρήση. Τα Volumes μπορούν να διαγραφούν, αντιγραφούν ή ακόμα και να μεταφερθούν σε άλλο εικονικό μηχάνημα αν υπάρχει συμβατότητα.

Guest OS Preferences, είναι οι επιλογές που έχουμε για το λειτουργικό σύστημα το οποίο θα φιλοξενηθεί από την πλατφόρμα νέφους της. Για παράδειγμα, μπορεί να θέλουμε να χρησιμοποιήσουμε Mac OS λογισμικό, αλλά αυτό είναι δεν μπορεί να γίνει διότι καμιά από της πλατφόρμες της δεν μπορεί να φιλοξενηθεί σε αυτό το λογισμικό.

Στην συνέχεια αναφέρεται το χαρακτηριστικό του **Host Maintenance with Live Migration.** Αυτό το χαρακτηριστικό είναι πολύ σημαντικό ώστε το σύστημα να λειτουργεί σε περίπτωση βλάβης ή συντήρησης. Live Migration στα συστήματα νέφους είναι η μεταφορά κάποιου εικονικού μηχανήματος σε διαφορετικό φυσικό μηχάνημα. Αυτή μεταφορά δεν θα πρέπει να επηρεάσει την λειτουργία και την παροχή υπηρεσιών του. Μαζί με την μεταφορά του VM, μεταφέρονται και τα χαρακτηριστικά του της η μνήμη, η χωρητικότητα, η δικτυακή συνδεσιμότητα κτλ.

Ένα εξίσου πολύ σημαντικό χαρακτηριστικό είναι το γεγονός αν το προϊόν παρέχεται **δωρεάν** ή όχι. Η ελεύθερη ή μη διάθεση της κάθε πλατφόρμας δεν σχετίζεται με την ποιότητα. Υπάρχουν πλατφόρμες που διανέμονται και χρησιμοποιούνται δωρεάν και είναι καλύτερές ή ίσες με της πλατφόρμες που πρέπει ο χρήστης να πληρώσει ώστε να μπορέσει να της χρησιμοποιήσει.

Προτελευταίο χαρακτηριστικό είναι το **High Availability**. Το σύστημα θα πρέπει να είναι διαθέσιμο όσο το δυνατό περισσότερο. Για παράδειγμα, σε περίπτωση βλάβης θα πρέπει να αναπτυχθούν όλοι εκείνοι οι μηχανισμοί οι οποίοι

θα την αναλάβουν και θα την αποκαταστήσουν. Λάθη και προβλήματα μπορούν να προκύψουν σε οποιοδήποτε φάσμα της συστήματος νέφους. Ακραίες περιπτώσεις της φυσικές καταστροφές που θα επηρεάσουν το φυσικό μηχάνημα αλλά και προβλήματα εντός του νέφους, της μια αναπάντεχη επανεκκίνηση της μηχανήματος ή κάποιο δικτυακό λάθος. Σε οποιαδήποτε περίπτωση θα πρέπει να είμαστε έτοιμοι έτσι ώστε τα επίπεδα της διαθεσιμότητας να παραμείνουν υψηλά. Μέθοδοι αντιμετώπισης τέτοιων περιπτώσεων μπορεί να είναι τα συχνά backup των δεδομένων, η παρακολούθηση και η ρύθμιση ειδοποιήσεων σε περίπτωση που κάτι δεν λειτουργεί σωστά κτλ.

Τέλος έχουμε το χαρακτηριστικό που λέγεται **Implementation Complexity.** Είναι πολύ λογικό να θέλουμε να χρησιμοποιήσουμε μια πλατφόρμα η οποία να είναι εύκολη στην σχεδίαση και στην υλοποίηση της. Στον πίνακα της παραπάνω αναφέρεται ως πολυπλοκότητα . Αξίζει να σημειωθεί ότι οι πλατφόρμες με την ένδειξη μείον(-) έχουνε μικρότερη πολυπλοκότητα. Γίνεται κατανοητό, λοιπόν, ότι αναπτύσσονται ευκολότερα.

Σημειώσεις:

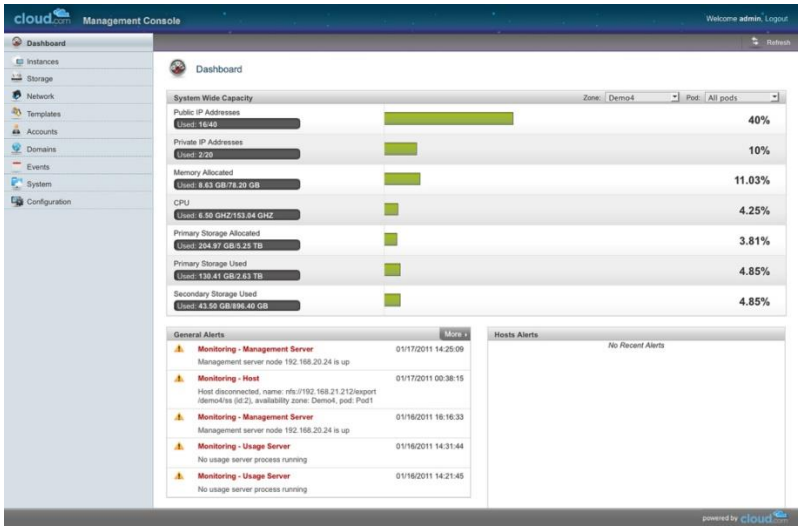
1. Το CloudStack δεν παρέχει της χρήστες του Γραφική Διεπαφή για την εφαρμογή Active Directory. Βέβαια, το API του CloudStack δίνει την δυνατότητα αυτή. Για αυτόν τον λόγο έχουν γραφτεί scripts από έμπειρους χρήστες έτσι ώστε να κάνουν την εφαρμογή του AD εύκολη.
2. Η κονσόλα διαχείρισης του Eucalyptus δεν διατίθεται στην δωρεάν έκδοση του
3. Η πρόσβαση μέσω Web στην εικονική μηχανή του Eucalyptus είναι διαθέσιμη μόνο με τον KVM hypervisor
4. Το είδος του λειτουργικού συστήματος που θα χρησιμοποιήσουμε για να φιλοξενήσει το νέφος της εξαρτάται από τον hypervisor, ειδικά, το Eucalyptus υποστηρίζει μόνο λειτουργικά συστήματα Linux.
5. Της έχουμε αναφέρει ήδη, το Eucalyptus διαθέτει δωρεάν και εμπορική έκδοση.
6. Η πολύπλοκη εγκατάσταση και παραμετροποίηση μέσω της εκάστοτε πλατφόρμας χωρίς την χρήση μιας διεπαφής, φιλικής της τον χρήστη, αποτελεί πρόκληση για αυτόν. Σε κάθε περίπτωση, η χρήση μιας διεπαφής

κάνει την εγκατάσταση ευκολότερη με αποτέλεσμα το Openstack και το Cloudstack να το πετυχαίνουν αυτό.

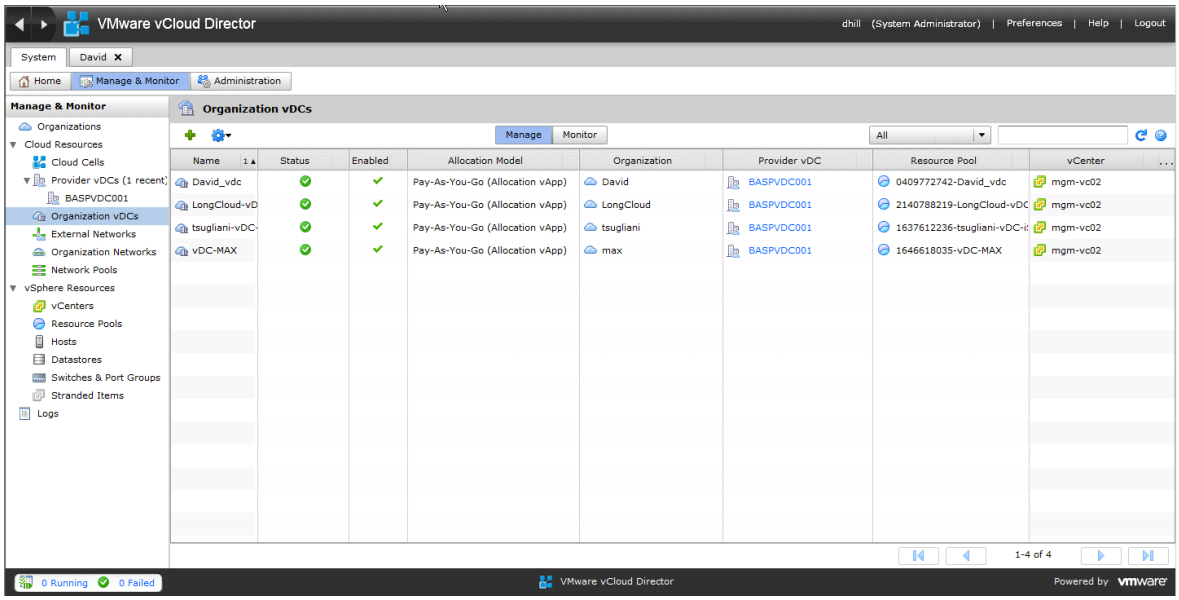
Στο σημείο αυτό και αφού είδαμε την ερμηνεία των χαρακτηριστικών του ανωτέρω πίνακα καθώς και της σημειώσεις και παρατηρήσεις για μερικά από αυτά, θα συγκρίνουμε της τέσσερις πλατφόρμες ανάλογα με το κάθε χαρακτηριστικό έτσι ώστε να φτάσουμε σε ένα ασφαλές και λογικό συμπέρασμα για την επιλογή του κατάλληλου λογισμικού που θα της βοηθήσει στην ανάπτυξη του εγχειρήματός της.

Αρχικά με την εφαρμογή του Active Directory, παρατηρούμε ότι μόνο το Openstack και το vCloud την παρέχουν μιας και το CloudStack δεν την παρέχει άμεσα αλλά μέσω της κοινότητάς των χρηστών του και το Eucalyptus δεν την καλύπτει. Η απουσία του AD αποτελεί μειονέκτημα στον τομέα της ταυτοποίησης και γενικά της ασφάλειας του νέφους.

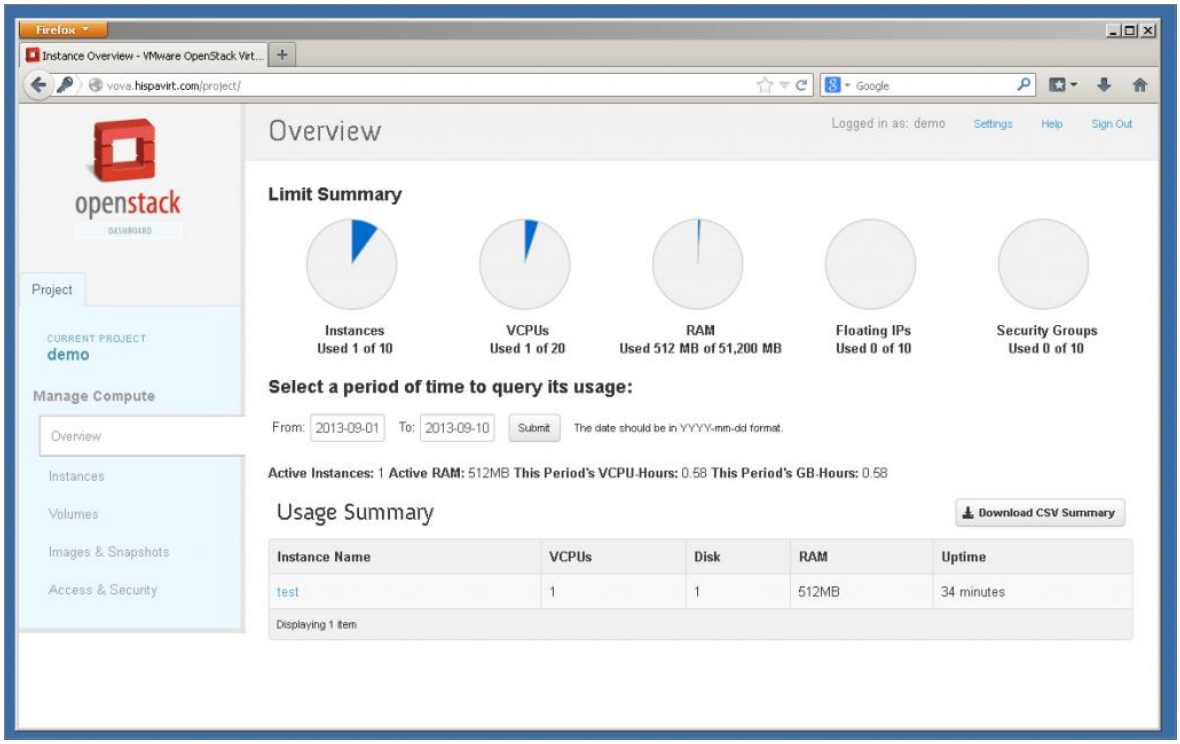
Η χρησιμοποίηση μια κονσόλας διαχείρισης μπορεί να κάνει αυτήν αλλά και τη συντήρηση του νέφους πολύ πιο εύκολη και γρήγορή. Έτσι μιας και το Eucalyptus δεν την παρέχει στην δωρεάν έκδοση το καθιστά λιγότερο προσιτό στην χρήση αλλά και στην επιλογή σε σχέση με τα άλλα λογισμικά που παρέχουν το καθένα την δικιά του κονσόλα.



3.17 CloudStack Management Console



3.18 vCloud Director Management Console



3.19 Openstack Management Console

Επόμενο χαρακτηριστικό είναι αυτό της πρόσβασης σε εικονικές μηχανές μέσω Web. Όπως βλέπουμε και στον πίνακα όλες οι πλατφόρμες το υποστηρίζουν με εξαίρεση το Eucalyptus που έχει μια δικλείδα και το υποστηρίζει μόνο με την χρήση του KVM Hypervisor. Η Web πρόσβαση είναι πολύ σημαντική για το σύστημα που θέλουμε να αναπτύξουμε διότι η πρόσβαση στα λειτουργικά συστήματα των εικονικών μηχανών γίνεται μέσω Web.

Αμέσως μετά έχουμε το χαρακτηριστικό της παροχής του προσωπικού API της κάθε πλατφόρμας. Είναι λογικό να παρέχεται απ' όλες τις πλατφόρμες διότι σε αντίθετη περίπτωση δεν θα ήταν δυνατή η ανάπτυξη ενός συστήματος νέφους μιας και χρησιμοποιούμε τα εργαλεία και τις βιβλιοθήκες του API για να την πετύχουμε.

Το Multi-Role Support είναι χαρακτηριστικό και των τεσσάρων λογισμικών. Είναι βασικό να υπάρχει πολλαπλότητα ρόλων διότι δεν μπορεί να υπάρξει σύστημα που να έχει έναν μοναδικό ρόλο. Για παράδειγμα ας πάρουμε την περίπτωση που σε ένα σύστημα όλοι θα έχουν τον ρόλο του διαχειριστή ανεξαρτήτως γνώσεων και εμπειρίας πάνω στην διαχείριση και την συντήρηση του συστήματος. Αυτό μπορεί να έχει καταστροφικές συνέπειες, καθώς κάποιος με μηδαμινή γνώση και δυνατότητες θα μπορούσε να κάνει ενέργειες που δεν θα ήταν της δικής του δικαιοδοσίας με αποτέλεσμα να διακινδυνεύεται ακόμα και η ύπαρξη του συστήματος.

Ακόμα ένα χαρακτηριστικό που υποστηρίζεται από όλες τις επιλογές μας είναι τα VLANs. Με την χρήση VLANS μπορούμε να ασφαλίσουμε καλύτερα το περιβάλλον μας μιας και μπορούμε να περιορίσουμε και να ελέγξουμε την κίνηση προς αλλά και από το εκάστοτε VLAN. Για το δικό μας σύστημα ένα VLAN θα μπορεί να φιλοξενεί τα διάφορα εικονικά μηχανήματα που δημιουργούνται προστατεύοντας τα από κινδύνους.

Συνεχίζοντας, ένα ακόμα γνώρισμα όλων των λογισμικών είναι η πολλαπλότητα στην υποστήριξη διαφορετικών Hypervisors. Αυτό δίνει την δυνατότητα επιλογής του hypervisor της επιλογής του καθενός με υποκειμενικά κριτήρια για τον καθένα. Βέβαια στην περίπτωση του vCloud, η VMWare που είναι η εταιρία κατασκευής του, προτείνει την χρήση του δικού της Hypervisor που λέγεται VSphere.

Αμέσως επόμενο χαρακτηριστικό βάσει του οποίου θα συγκριθούν οι πλατφόρμες είναι η διαδικασία της δημιουργίας Templates. Εξαίρεση αποτελεί το Eucalyptus το οποίο δεν δίνει την δυνατότητα δημιουργίας templates. Η χρήση των templates μπορεί να φανεί χρήσιμη στο δικό μας εγχείρημα καθώς μπορούμε να διαμορφώνουμε τις εικόνες των λειτουργικών συστημάτων σύμφωνα με τις

απαιτήσεις της κάθε περίπτωσης και στην συνέχεια να τις μετατρέπουμε σε templates διευκολύνοντας την μετέπειτα διαχείριση και συντήρηση τους.

Συνεχίζοντας, όλες οι πλατφόρμες υποστηρίζουν την δημιουργία Snapshots. Με αυτόν τον τρόπο επιταχύνεται και απλοποιείται η διαδικασία χρήσης, διαχείρισης και μεταφοράς των εικονικών μηχανών ακόμα και σε περιπτώσεις βλάβης ή καταστροφής των φυσικού εξοπλισμού.

Ένα ακόμα γνώρισμα που το συναντάμε και στις τέσσερις επιλογές μας είναι το Resource Over Provisioning Limits. Αυτό το χαρακτηριστικό είναι πολύ σημαντικό στην περίπτωση του εγχειρήματος μας καθώς μπορούμε να υπολογίσουμε πόσα εικονικά μηχανήματα μπορεί να υποστηρίξει ο εξοπλισμός μας. Έτσι με αυτόν τον τρόπο αν ο τωρινός μας εξοπλισμός δεν είναι αρκετός μπορούμε να υπολογίσουμε ανάλογα τους πόρους που θα χρειαστούν, τις περαιτέρω προσθήκες που είναι αναγκαίες για να υλοποιηθεί το εγχείρημα μας.

Alerts και Notifications είναι ένα χαρακτηριστικό που υποστηρίζεται από όλα τα λογισμικά πλην του Eucalyptus. Αν και η υποστήριξη του δεν είναι πολύ σημαντική, παρέχει επιπλέον βοήθεια στους υπεύθυνους του περιβάλλοντος ώστε να διορθώσουν τυχόν προβλήματα που θα προκύψουν.

Ακολουθεί το γνώρισμα της υποστήριξης των Volumes , όπου και τα τέσσερα λογισμικά νέφους το παρέχουν. Είναι θετικό να υπάρχει η υποστήριξη αυτή διότι μας δίνεται μεγαλύτερη ευχέρεια στην λειτουργία του περιβάλλοντος μας με αποθηκευτική ισχύ ή οποία δεν χάνεται και μπορεί να τροποποιηθεί ανάλογα με τις ανάγκες.

Αμέσως μετά έρχεται το Guest OS Preferences χαρακτηριστικό. Όπως μπορούμε να δούμε και στον πίνακα, όλα τα λογισμικά της επιλογής μας έχουν την δυνατότητα αυτήν, εκτός από το Eucalyptus που σύμφωνα με την 4η σημείωση του πίνακα μπορεί να φιλοξενήσει μόνο Linux OS. Στην περίπτωση του Lab as a Service αυτό δεν είναι αρκετό καθώς θέλουμε το σύστημα μας να υποστηρίζει και Windows OS μιας και χρησιμοποιείται κατά κόρον από τους φοιτητές του τμήματος.

Επόμενο χαρακτηριστικό σύμφωνα με τον πίνακα είναι το Host Maintenance with Live Migration. Υποστηρίζεται από το CloudStack και το vCloud Director και όχι από το Openstack και το Eucalyptus. Αρκετά σημαντικό για την

σωστή λειτουργία του νέφους αν θέλουμε η διαθεσιμότητα του συστήματος μας να είναι σε υψηλότατα επίπεδα.

Ακολουθεί το χαρακτηριστικό που είναι ένα από τα πιο σημαντικά στην επιλογή της πλατφόρμας ανάπτυξης του Lab as a Service αλλά και γενικά ένα από τα χαρακτηριστικά που επικεντρώνονται οι εταιρίες για να καταστρώσουν και να σχεδιάσουν τα δικά τους συστήματα. Το χαρακτηριστικό είναι η δωρεάν χρήση του λογισμικού. Μόνο το Openstack και το CloudStack παρέχουν δωρεάν τις υπηρεσίες τους, σε αντίθεση με το vCloud που για να χρησιμοποιήσει κάποιος τις υπηρεσίες τους πρέπει να αγοράσει το προϊόν και με το Eucalyptus που παρέχει μερικές μόνο υπηρεσίες του δωρεάν και για να χρησιμοποιήσεις πλήρως την πλατφόρμα πρέπει να κατέχεις την εμπορική έκδοση που παρέχεται επί πληρωμή.

Προτελευταίο χαρακτηριστικό αποτελεί το High Availability Cloud Component με την υποστήριξή του να παρέχεται και πάλι από όλες τις πλατφόρμες. Χαρακτηριστικό που αποτελεί βασικό για την σωστή και διαρκή λειτουργία του νέφους και σκοπός του είναι να διασφαλίζει δύο κύρια ζητήματα, την μη απώλεια δεδομένων και να ελαχιστοποιεί τον χρόνο που το σύστημα δεν θα λειτουργεί σε περίπτωση που συμβεί κάτι.

Φτάνουμε και στο τελευταίο γνώρισμα το οποίο είναι το Implementation Complexity. Με το Eucalyptus και το Openstack να μην το υποστηρίζουν και συμπερασματικά να τα κάνουν πιο αρεστά από το CloudStack και το vCloud Director, καθώς με την χρήση των δύο πρώτων θα είναι πιο εύκολη η υλοποίηση του νέφους.

Επίλογος

Αρχικά είδαμε και τα χαρακτηριστικά των CloudStack, Eucalyptus,vCloud Director και Openstack, μαζί με την ασφάλεια που παρέχεται από το καθένα και είναι αναγκαία για την σωστή και συνεχή λειτουργία του νέφους. Στην συνέχεια παραθέσαμε έναν πίνακα για να μπορούμε να έχουμε μια γενική εικόνα με κάποια βασικά χαρακτηριστικά που πρέπει να έχει η πλατφόρμα που θα επιλέξουμε ώστε να αναπτύξουμε το Lab as a Service. Συγκρίνοντας λοιπόν τα συστήματα μας με τα χαρακτηριστικά αυτά καταλήξαμε στην επιλογή του Openstack για την υλοποίηση μας. Βασικές συνιστώσες στην επιλογή μας αυτήν ήταν το γεγονός ότι παρέχεται δωρεάν. Επίσης πολύ σημαντικό ήταν το γεγονός ότι έχει την

μεγαλύτερη κοινότητα χρηστών και ότι έχει την μεγαλύτερη πρόοδο ανάπτυξης σε σχέση με όλα τα υπόλοιπα. Παράλληλα είναι αυτό που μας παρέχει την μεγαλύτερη ελευθερία στην επιλογή hypervisors , λειτουργικών συστημάτων κτλ. Στο κεφάλαιο που ακολουθεί θα δούμε με λεπτομέρεια κάποια πράγματα παραπάνω για το Openstack, θα αναλύσουμε τις διαφορετικές υπηρεσίες που θα χρησιμοποιήσουμε για την ανάπτυξη του Lab as a Service καθώς και το πώς αναπτύξαμε την κάθε υπηρεσία ξεχωριστά που ο συνδυασμός τους μας δίνει το επιθυμητό αποτέλεσμα.

4 Περιγραφή του Lab as a Service

Εισαγωγή

Το Lab as a Service (LaaS) όπως προδίδει και το όνομα του, έχει ως στόχο να παρέχει στους χρήστες του μέσω του υπολογιστικού νέφους, μια υπηρεσία διανομής εικονικού εργαστηριακού περιβάλλοντος. Ως χρήστες στην προκείμενη περίπτωση νοούνται τόσο οι φοιτητές του τμήματος Πληροφορικής (οι οποίοι θα είναι απλοί χρήστες στο σύνολό τους) αλλά και οι καθηγητές (οι οποίοι μπορεί να είναι χρήστες ή/και διαχειριστές συστήματος παράλληλα).

4.1 Περιγραφή του μοντέλου Lab-as-a-Service (LaaS)

Πιο συγκεκριμένα η φιλοσοφία γύρω από το LaaS είναι η ανεξαρτητοποίηση από τον φυσικό χώρο του εργαστηρίου καθώς και από το υλικό από το οποίο αυτός συνοδεύεται (ηλεκτρονικοί υπολογιστές/λειτουργικά συστήματα). Ιδανικά θα θέλαμε οποιοσδήποτε φοιτητής έχει στην διάθεση του έναν ηλεκτρονικό υπολογιστή καθώς και πρόσβαση στο ιδιωτικό δίκτυο του τμήματος να είναι σε θέση να εισέρχεται στην υπηρεσία διανομής εικονικών εργαστηρίων του νέφους (Lab as a service) και να χρησιμοποιεί προσωποποιημένες εικόνες λειτουργικών συστημάτων οι οποίες προσομοιώνουν το εργαστηριακό περιβάλλον.

Τα εικονικά λειτουργικά αυτά συστήματα θα είναι πανομοιότυπα με αυτά που ήδη τρέχουν στους υπάρχοντες σταθερούς ηλεκτρονικούς υπολογιστές κάθε εργαστηρίου με εγκατεστημένα προγράμματα και λειτουργίες που απαιτούνται για τα διάφορα μαθήματα. Σε κάθε φοιτητή θα παρέχεται επίσης, ένας προσωποποιημένος αποθηκευτικός χώρος πάνω στο νέφος στον οποίο θα μπορεί να αποθηκεύει αυτά τα εικονικά λειτουργικά συστήματα (Windows, Unix κτλ.) σώζοντας έτσι κατά έναν τρόπο την πρόοδο του στο εργαστηριακό μέρος οποιουδήποτε μαθήματος ενώ παράλληλα έχει πρόσβαση σε αυτά από οποιοδήποτε μέρος υπάρχει ένας ηλεκτρονικός υπολογιστής με δυνατότητα virtualization και πρόσβαση στο εσωτερικό δίκτυο του τμήματος.

Από όλα τα παραπάνω προκύπτει η ανάγκη να επεξηγηθεί πως το εσωτερικό δίκτυο του τμήματος είναι άμεσα προσβάσιμο μόνο μέσω κάποιας φυσικής συσκευής σύνδεσης στο ίδιο το κτήριο αλλά επίσης και μέσω της

υπηρεσίας VPN που παρέχεται κάτι που θα καθιστά εύκολη την απομακρυσμένη πρόσβαση στο LaaS. Η είσοδος των φοιτητών θα γίνεται μέσω μιας φόρμας με ένα προσωποποιημένο μοναδικό «όνομα χρήστη» και έναν «κωδικό πρόσβασης» αν και μελλοντικά ως υποδομή μπορεί να συνδυαστεί με τις υπάρχουσες υπηρεσίες του Τμήματος κάτι που σημαίνει πως οι φοιτητές θα είναι δυνατό να έχουν πρόσβαση στο νέφος και την υπηρεσία LaaS με το email και τον κωδικό που τους έχει δοθεί ήδη από την σχολή.

4.2 Υφιστάμενες Lab-as-a-Service λύσεις

Το μοντέλο Lab-as-a-Service που περιγράφεται σε αυτή την εργασία είναι αρκετά καινοτόμο και θα μπορούσε να αποτελεί μια σίγουρη λύση για πολλά ιδρύματα στο μέλλον καθώς θα μπορεί να παρέχει τις ίδιες υπηρεσίες με αυτές που υφίστανται ως τώρα , αλλά και παράλληλα να εξοικονομεί πόρους τόσο σε υλικό όσο και σε συντήρηση.

Με μια γρήγορη αναζήτηση στο διαδίκτυο μπορεί κανείς να βρει πολύ εύκολα τόσο γενικευμένες παροχές υπηρεσιών τύπου LaaS παγκοσμίως, όσο και λύσεις που αποτελούν ίδια υλοποίηση με αυτήν του LaaS και έχουν δημιουργηθεί και υιοθετηθεί από πανεπιστημιακά ιδρύματα.

Ξεκινώντας, θα κάνουμε μια αναφορά στις λύσεις που είναι παρόμοιες με το Laas.Πολλές εταιρίες πληροφορικής δημιούργησαν και παρέχουν υπηρεσίες παρόμοιες και πιο γενικές του Lab-as-a-Service μοντέλου. Έτσι, η λέξη «Lab» παίρνει μια πιο ευρεία έννοια και μπορεί να είναι ένα επιστημονικό εργαστήριο, ένα ερευνητικό εργαστήριο, ένα εργαστήριο πειραματικών δοκιμών και αποτελεσμάτων κ.ο.κ. Τα πακέτα προσφοράς αυτών των εταιρειών για αυτό το λόγο είναι επίσης πιο γενικά και προσφέρουν πλούσιο υλικό, και πολλές και διαφορετικές λειτουργίες σε ότι αφορά τους τελικούς χρήστες. Πιο απλοϊκά αυτό που προσφέρεται από αυτές τις εταιρείες είναι μια δυνατότητα μετανάστευσης οποιουδήποτε εργαστηριακού περιβάλλοντος σε υποδομές cloud αλλά δεν συγκεκριμενοποιούνται σε ένα τύπο εργαστηρίου όπως το δικό μας μοντέλο. Τέτοιες εταιρείες είναι οι:

- Qualisystems: http://www.qualisystems.com/solutions/lab-management/lab-as-a-service-laas/
- Virsoft: http://www.virsoft.net/Pages/Labs.aspx

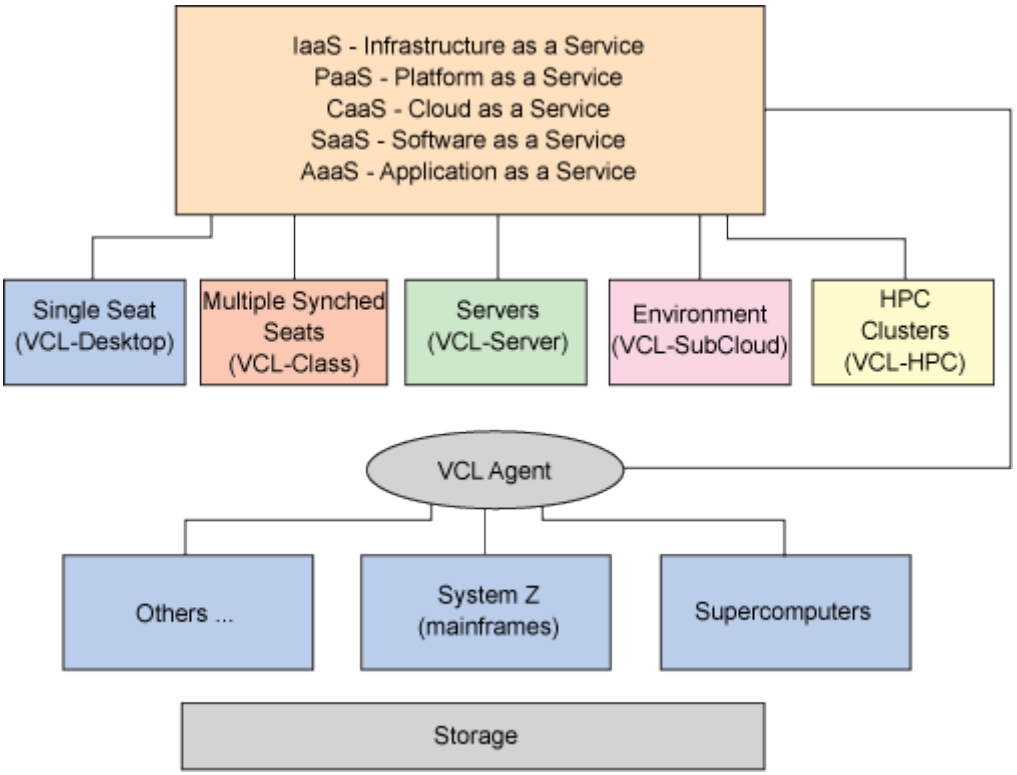
- WWT: http://www2.wwt.com/content/lab-service
- CISCO: http://solutionpartner.cisco.com/web/partner/laas

Στην συνέχεια, και αφού πρώτα αναφέραμε κάποιες λύσεις που είναι παρόμοιες με την υλοποίηση μας, θα θέλαμε να αναφερθούμε εκτενώς σε μια λύση που είναι εφάμιλλη, αν όχι η ίδια με αυτήν του LaaS και έχει ακαδημαϊκό σκοπό και στόχο. Η λύση αυτή λέγεται **Virtual Computing Lab**.

4.3 Virtual Computing Lab

Το **Virtual Computing Lab (VCL)**[30] είναι μια ιδέα βασίζεται πάνω στην τεχνολογία του υπολογιστικού νέφους. Αναπτύχθηκε στο North Carolina State University (NCSU) μέσω της συνεργασίας του τμήματος μηχανικών του κολλεγίου και του τμήματος της IBM που είναι αρμόδιο για την εγχειρήματα εικονικών υπολογιστικών συστημάτων (Virtual Computing). Σκοπός ήταν η αντιμετώπιση του ολοένα αυξανόμενου συνόλου των υπολογιστικών αναγκών και των απαιτήσεων των χρηστών για το πανεπιστήμιο. Αυτό το σύστημα μπορεί να προσφέρει λύσεις χρήστη που απαιτούνται για την ποικιλία των περιβαλλόντων υπηρεσιών οποτεδήποτε και οπουδήποτε με τη ζήτηση / κράτηση.

Το VCL αποτελεί έναν συνδυασμό των τριών βασικών αρχιτεκτονικών του υπολογιστικού νέφους (IaaS, PaaS, SaaS).



4.1 Οι Υπηρεσίες του νέφους VCL

Όσον αφορά το IaaS το VCL προσφέρει μια διαφορετική υποδομή σε έναν τόπο. Παρέχει μια πλατφόρμα εικονικού περιβάλλοντος σε πανεπιστημιακά ιδρύματα. Έτσι οι φοιτητές, χρησιμοποιώντας το VCL δεν είναι αναγκασμένοι να εγκαταστήσουν κάποια φυσική υποδομή για την ανάγκη των εργασιών τους.

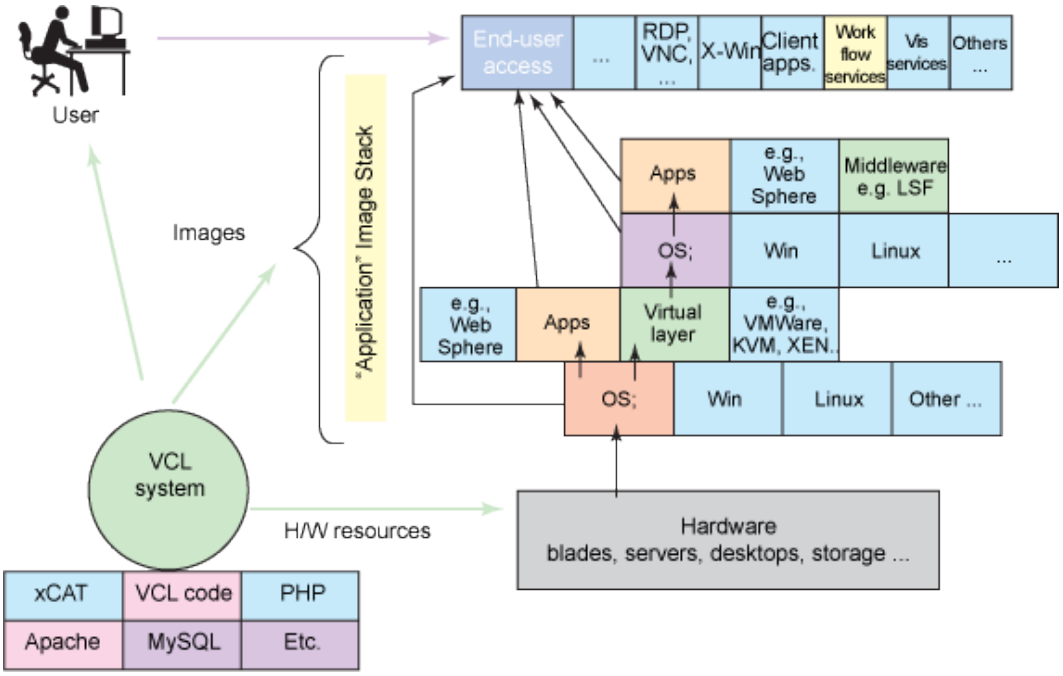
Το VCL παρέχει υπηρεσίες υπολογιστικού περιβάλλοντος(Φυσικές μηχανές, εικονικές μηχανές και εικονοποίηση σε επίπεδο λειτουργικών συστημάτων), δικτύου και αποθηκευτικού χώρου. Επίσης, ο VCL manager διαχειρίζεται κατάλληλα τους διαθέσιμους πόρους του υλικού και παρέχει τις κατάλληλες τεχνικές εικονοποίησης ώστε οι διαθέσιμες εικόνες να είναι λειτουργικές προς τους φοιτητές. Τέλος , σκοπός των υπηρεσιών του VCL είναι ο έλεγχος των πόρων σε επίπεδο πλατφόρμας.

Συνεχίζοντας , το VCL υιοθετεί και χαρακτηριστικά του PaaS μοντέλου. Αυτό συμβαίνει διότι , οι φοιτητές χρησιμοποιώντας το VCL δεν χρειάζεται να εγκαταστήσουν καμία συγκεκριμένη υπηρεσία ή κάποια βάση δεδομένων στο φυσικό μηχάνημα τους. Το VCL παρέχει τις εικόνες στους φοιτητές οι οποίοι στην συνέχεια μπορούν να επιλέξουν οποιαδήποτε εικόνα και να την χρησιμοποιήσουν πάνω σε ένα μηχάνημα που βρίσκεται στο νέφος.

Υπάρχουν έτοιμες λύσεις με εικόνες για Java , PHP καθώς και για .NET. όπως επίσης και αποθηκευτικός χώρος για αρχεία και για βάσεις δεδομένων που μπορεί να χρησιμοποιεί ο φοιτητής.

Τέλος , για την εφαρμογή του SaaS μοντέλου, το VCL επιτρέπει οποιαδήποτε των τωρινών SaaS λύσεων όπως τα VMWare, XEN, MS Virtual Server, Virtuoso και Citrix. Επίσης επιτρέπει οποιαδήποτε επιλογή παράδοσης μέσω VNC ή RDP.

Το VCL χαρακτηρίζεται από μία αρχιτεκτονική υψηλού επιπέδου. Η αρχιτεκτονική αυτή προορίζεται κυρίως για τον σχεδιασμό και την διαμόρφωση ενός συστήματος νέφους που εξυπηρετεί τόσο τις εκπαιδευτικές όσο και τις ερευνητικές αποστολές του πανεπιστημίου με έναν πολύ οικονομικό και αποδοτικό τρόπο. Το VCL προσφέρει μια σειρά από υπηρεσίες και λειτουργίες που δρουν καλά πάνω στο σύστημα του νέφους και καλύπτουν όλες τις προσδοκίες. Αυτή η αρχιτεκτονική αποτελείται από κάποια κύρια συστατικά όπως μπορούμε να δούμε και στο Σχήμα 2 που ακολουθεί.



4.2 Η Φυσική Αρχιτεκτονική του VLC

Αρχικά, ο χρήστης αποκτά πρόσβαση στο VCL μέσω ενός web interface για να επιλέξει τον επιθυμητό συνδυασμό της εφαρμογής από ένα μενού όπως φαίνεται και στο Σχήμα 3 παρακάτω. Αν μια συγκεκριμένη εικόνα χρήστη δεν είναι

διαθέσιμη, τότε ένας εξουσιοδοτημένος χρήστης μπορεί να έχει την ευελιξία να την κατασκευάσει με την βοήθεια των διαφόρων συστατικών που υπάρχουν στην βιβλιοθήκη του VCL. Στην συνέχεια, το λογισμικό του VCL manager αντιστοιχίζει αυτό το αίτημα του χρήστη σε πιθανόν διαθέσιμες εικόνες και πόρους του υλικού και το διευθετεί είτε προς άμεση χρήση είτε προς το μέλλον.

4.3 Δημιουργία εικόνας μέσω web interface

Ο τρόπος πρόσβασης στους πόρους και τις υπηρεσίες του συστήματος εξαρτάται από τις επιλογές που προσφέρει η εκάστοτε υπηρεσία. Ένα παράδειγμα φαίνεται στο Σχήμα 4. Μπορεί να κυμαίνεται από RDP ή VNC πρόσβαση σε μια απομακρυσμένη επιφάνεια εργασίας, σε μια SSH σύνδεση ή X-Win σε ένα περιβάλλον Linux, σε web-based πρόσβαση και τέλος σε proxy σύνδεση σε ένα υπολογιστικό cluster.

4.4 Πρόσβαση μέσω SSH σε περιβάλλον Red Hat Linux

Αναφέραμε προηγουμένως τον VCL manager. Σκοπός του είναι ο έλεγχος των διαφόρων περιβαλλόντων, η διαχείριση των ηλεκτρονικών υπολογιστών και των διαθέσιμων εικόνων. Το λογισμικό του VCL manager περιλαμβάνει επιγραμματικά τα ακόλουθα προϊόντα:

- **IBM xCAT and VM loader.** Το Extreme Cluster Administration Toolkit είναι μια συλλογή από scripts για την κατασκευή, διαμόρφωση και διαχείριση των

Linux clusters. Το VCL χρησιμοποιεί το xCAT για να φορτώσει το image στον server.

- **VCL Middle Layer Demon Service (VCLD).** Το βασικό μέρος του VCL Manager είναι μία υπηρεσία βασισμένη στην Perl, η οποία χρησιμοποιείται τόσο για την πρόβλεψη όσο και για την ανάπτυξη.
- **Ένας Web Server ανοιχτού κώδικα (Apache).** Αυτή η web εφαρμογή βασισμένη στην PHP είναι η καρδιά του VCL και παρέχει διάφορα εργαλεία για την διαχείριση και ρύθμιση όλων των πόρων του VCL.
- **Βάση Δεδομένων ανοιχτού κώδικα (MySQL).** Χρησιμοποιείται για την παρακολούθηση της κατάστασης όλων των servers και για την διατήρηση των πληροφοριών κάθε εικόνας.

Το VCL παρέχει και κατάλληλη ασφάλεια προς τους χρήστες. Αρχικά θα θέλαμε να επισημάνουμε είναι δύσκολο να ορίσουμε την ασφάλεια στο πλαίσιο ενός νέφους. Το απόλυτο επίπεδο των μέτρων ασφαλείας που απαιτούνται για κάθε κατανεμημένο σύστημα προέρχεται τόσο από επαλήθευση ταυτότητας όσο και από την εξουσιοδότηση των υπηρεσιών. Έτσι, το VCL εφαρμόζει το ακόλουθα επίπεδα ασφαλείας στο σύστημα:

- **Αυθεντικοποίησης LDAP**.
- **Αυθεντικοποίησης βάσει του επιπέδου του περιβάλλοντος.**

Βέβαια, εκτός από τις παραπάνω, εάν ένας χρήστης έχει το δικαίωμα της δημιουργίας εικόνων τότε, το VCL κλειδώνει την IP του περιβάλλοντος και την IP του χρήστη χρησιμοποιώντας Firewall σε επίπεδο λειτουργικού συστήματος.

Το VCL χρησιμοποιείται από πάρα πολλά πανεπιστημιακά ιδρύματα. Με μία πολύ απλή και γρήγορη αναζήτηση μπορούμε να βρούμε ποια είναι τα ιδρύματα αυτά. Χαρακτηριστικά, μερικά από αυτά είναι τα πανεπιστήμια στην Washington, στο Duke, στο Maryland, στην North Carolina και σε άλλα πολλά ιδρύματα τα οποία χρησιμοποιούν τις υπηρεσίες του VCL.

Συνοψίζοντας, μπορούμε να συμπεράνουμε ότι το VCL είναι ένα Web-based σύστημα ανοιχτού κώδικα που χρησιμοποιείται για την παροχή απομακρυσμένης πρόσβασης σε ένα ειδικό περιβάλλον του υπολογιστή ενός χρήστη. Το VCL σύννεφο παρέχει εξαιρετική υπολογιστική ισχύ μέσω ενός

μοναδικού ανοιχτού λογισμικού για να τρέξει και να φιλοξενήσει όλα τα πανεπιστημιακά σχέδια και προγράμματα εκμάθησης ενός ιδρύματος.

Επίλογος

Μετά από την περιγραφή του Lab as a Service ως ιδέα και ως φιλοσοφία που αναλύθηκε σε αυτό το κεφάλαιο, όπως επίσης και των διαφόρων λογισμικών νέφους που συγκρίθηκαν στο προηγούμενο κεφάλαιο, λάβαμε την απόφαση να αναπτύξουμε το δικό μας Lab as a Service περιβάλλον για το Τμήμα Πληροφορικής του ΤΕΙ Θεσσαλονίκης χρησιμοποιώντας το Openstack λογισμικό νέφους με σκοπό των διαμοιρασμό εικόνων λειτουργικών συστημάτων προς τους φοιτητές του τμήματος.

5 Περιγραφή του τεχνολογικού περιβάλλοντος Openstack

Εισαγωγή

Στο προηγούμενο κεφάλαιο καταλήξαμε στο Openstack Cloud Software ως την ιδανικότερη λύση για το εγχείρημα του Lab As a Service . Έτσι λοιπόν στο κεφάλαιο αυτό θα δούμε αναλυτικά , Τι είναι το Openstack , ποια βασικά συστατικά του χρησιμοποιούνται σε όλα τα συστήματα νέφους που χρησιμοποιούν το Openstack καθώς και αυτά τα οποία εμείς χρησιμοποιήσαμε κατά την διάρκεια της δικής μας υλοποίησης, μαζί με τις εντολές και τις ρυθμίσεις για να γίνει το LaaS λειτουργικό . Επίσης θα δούμε πως όλα αυτά τα συστατικά συνδυάζουν τις δυνατότητες και ιδιότητες τους ώστε να μας δώσουν το τελικό αποτέλεσμα που επιζητούμε.

5.1 Η αρχή και η εξέλιξη του Openstack

Είναι ενδιαφέρον το πώς ξεκίνησε η ανάπτυξη του Openstack[31]. Το Openstack αρχικά ξεκίνησε από την NASA (http://www.nasa.gov/) και την Rackspace(http://www.rackspace.com/). Στην πραγματικότητα όμως το Openstack έκανε την πρώτη του εμφάνιση το 2010 ως ένα project της NASA σε συνεργασία με την Rackspace. Το εγχείρημα αυτό ξεκίνησε από την ιδέα της συνένωσης δύο προϊόντων τα οποία μέχρι τότε αναπτύσσονταν από τους δύο αυτούς οργανισμούς ξεχωριστά. Τόσο η NASA όσο και η Rackspace , δεν ήθελαν μια απλή ένωση των προϊόντων τους , αλλά να το αναπτύξουν και το προωθήσουν ως ένα πλήρως λειτουργικό λογισμικό ανοιχτού κώδικα για εγχειρήματα και συστήματα υπολογιστικού νέφους. Έτσι λοιπόν, συνδυάζοντας την υπολογιστική πλατφόρμα **Nebula** της NASA με την πλατφόρμα αρχείων νέφους της Rackspace δημιουργήθηκε η πρώτη έκδοση του Openstack.

Το Openstack δεν αφορά μόνο αυτό καθεαυτό το προϊόν . Πολύ σημαντική είναι επίσης και η διαχείριση του. Εδώ ακριβώς είναι το σημείο που λαμβάνει χώρα το Openstack ως οργανισμός , γνωστό ως **Openstack Foundation.** Ο όλος σχεδιασμός του Openstack Foundation έχει ως στόχο να μπορεί σημαντικές λειτουργίες που έχουν να κάνουν με την διαχείριση όχι μόνο την πλατφόρμας αλλά και ολόκληρου του οικοσυστήματος του νέφους. Όταν μιλάμε για οικοσύστημα

νέφους εννοούμε την πολυπλοκότητα ενός συστήματος , το οποίο αποτελείται από διαφορετικά συστατικά που αυτά με την σειρά τους συνεργάζονται και συνδυάζονται για να ενεργοποιηθούν οι εκάστοτε υπηρεσίες του υπολογιστικού νέφους αυτού. Όσο αφορά το Openstack Foundation , λοιπόν, δημιουργήθηκε για να παρέχει και να προωθεί συνεχώς την ελευθερία ,την διαφάνεια καθώς και την ανάπτυξη μέσω δραστηριοτήτων υπολογιστικού νέφους. Παράλληλα το οικοσύστημα του Openstack αφορά και την ανάπτυξη και την συνεργασία με άλλους εταίρους , οι οποίοι εμπλέκονται πλήρως σε αυτό το εγχείρημα.

Το Openstack Foundation αποτελείται από τρεις πολύ σημαντικές συνιστώσες όπως φαίνεται και στις εικόνες παρακάτω.

5.1 To Openstack Foundation

Αρχικά, η Τεχνική Επιτροπή ή αλλιώς **Technical Committee**. Η επιτροπή αυτή αποτελείται από 13 εκλεγμένα μέλη τα οποία έχουν και εξυπηρετούν τον ίδιο σκοπό. Ο σκοπός αυτός είναι η παροχή τεχνικής ηγεσίας για το Openstack ως σύνολο. Ευθύνη των 13 αυτών μελών είναι επίσης , η προώθηση και επιβολή των ιδεωδών του Openstack. Τέτοια ιδεώδη αποτελούν η διαφάνεια, η ποιότητα , ο ελεύθερος διαμοιρασμός του. Άλλες ευθύνες αποτελούν οι λήψεις αποφάσεων για θέματα που επηρεάζουν διάφορα άλλα προγράμματα , τεχνικά ζητήματα καθώς και θέματα γενικής εποπτείας.

Στην συνέχεια , υπάρχει , όπως και σε όλους τους οργανισμούς το διοικητικό συμβούλιο ή **Board of Directors.** Κύριο μέλημα του είναι η διοίκηση και η εποπτεία του οργανισμού καθώς και η διαχείριση του έμψυχου δυναμικού.

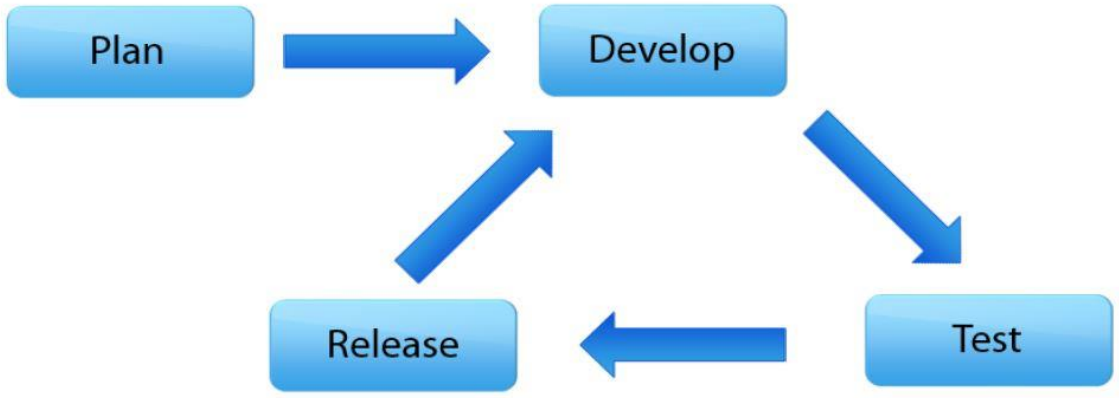
Τέλος , έχουμε την κοινότητα των χρηστών ή **User Committee.** Η κοινότητα των χρηστών είναι πολύ σημαντική για τον οργανισμό Openstack. Με την τόσο μεγάλη αύξηση των εφαρμογών που βρίσκει το Openstack καθώς και με την χρησιμοποίηση του από τρίτους , γίνεται όλο ένα και πιο σημαντικό η κοινότητα να πρέπει να δημιουργήσει υπηρεσίες αλλά και οδηγούς γύρω από το Openstack που θα έχουν ως αποτέλεσμα την εξέλιξη του προϊόντος. Κύρια αποστολή του User Committee είναι :

- Να παρουσιάζει στο Technical Committee και στο Directors Board τις απαιτήσεις των χρηστών αφού πρώτα τις αξιολογήσει.
- Να παρέχει καθοδήγηση και βοήθεια στις ομάδες ανάπτυξης λογισμικού σε περιπτώσεις που ζητείται η βοήθεια των χρηστών.
- Να παρακολουθεί τις διάφορες εφαρμογές και την χρήση του Openstack με απώτερο σκοπό να διαμοιραστούν οι εμπειρίες και οι προσωπικές ιστορίες των χρηστών.
- Να είναι σε επικοινωνία και συνεργασία με τις ομάδες χρηστών ανά τον κόσμο με σκοπό η κοινότητα του Openstack να είναι ενεργή και συνεχώς ενημερωμένη.

Το κομμάτι της κοινότητας που έχει ως μέλημα την ανάπτυξη κώδικα για υπηρεσίες του Openstack είναι τεράστιο. Υπάρχουν εκατοντάδες χιλιάδες προγραμματιστές που σαν ομάδες προσφέρουν καθημερινά στην ανάπτυξη του οργανισμού. Εκτός όμως από αυτούς υπάρχουν και ολόκληροι οργανισμοί όπως οι Rackspace , VMWare , RedHat , EMC που και αυτές με την σειρά τους προσφέρουν . Επιπροσθέτως , δεν υπάρχουν μόνο προγραμματιστές αλλά και άνθρωποι που έχουν αναλάβει την αρχειοθέτηση και την μηχανογράφηση όλου αυτού του όγκου πληροφορίας που καθημερινά διαμοιράζεται μέσω της κοινότητας.

Οι εκδόσεις του Openstack έχουν ενδιαφέρον στον τρόπο με τον οποίο γίνονται. Πρώτα , όσο αφορά την ονομασία τους, γίνεται με αλφαβητική σειρά. Αυτό τις καθιστά εύκολες στην απομνημόνευση βλέποντας μόνο το αρχικό γράμμα κάθε έκδοσης.

Όλες οι εκδόσεις ακολουθούν τον ίδιο κύκλο προτού δοθούν σε κυκλοφορία. Ο κύκλος αυτός φαίνεται στο παρακάτω σχεδιάγραμμα.



5.2 Ο κύκλος έκδοσης Openstack / Openstack Release Cycle

Όπως γίνεται αντιληπτό για να δοθεί μια έκδοση του Openstack στο κοινό υπάρχει μια ακολουθία που εφαρμόζεται. Αρχικά υπάρχει η συζήτηση και η σχεδίαση με το τι θα περιέχει η εκάστοτε έκδοση. Στην συνέχεια περνάμε στο στάδιο τις ανάπτυξης των αποφάσεων του πρώτου μέρους. Σε αυτό το σημείο αρχίζει και παίρνει μορφή και υπόσταση η έκδοση του Openstack. Αφότου τελειώσει η ανάπτυξη των υπηρεσιών , περνάμε στο στάδιο των δοκιμών. Σε αυτό το σημείο γίνονται οι απαραίτητες δοκιμές για την σωστή λειτουργία των καινούριων υπηρεσιών. Τέλος , αφότου γίνουν αυτές οι δοκιμές , γίνεται η έκδοση των ενημερώσεων και γίνονται διαθέσιμες προς τους χρήστες του Openstack. Αξίζει να σημειωθεί ότι , η ανάπτυξη και η δοκιμές δεν τελειώνουν με την έκδοση αλλά συνεχίζουν καθ' όλη την διάρκεια της ύπαρξης της εκδόσεως με απώτερο σκοπό την βελτίωση των υπηρεσιών που προσφέρει η κάθε έκδοση Openstack.

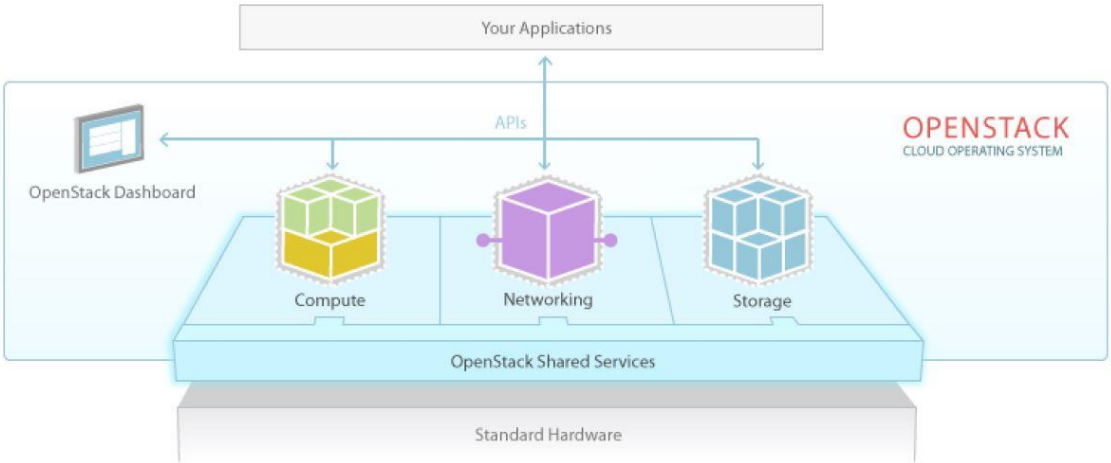
Ένα άλλο αξιοσημείωτο χαρακτηριστικό αποτελεί ο γρήγορος ρυθμός με τον οποίο οι εκδόσεις δίνονται σε κυκλοφορία. Όπως βλέπου και στην Εικόνα 5 μόλις σε διάστημα 4 χρόνων έχουν κυκλοφορήσει 10 διαφορετικές εκδόσεις και η καθεμία από αυτές έχει να προσφέρει κάτι καινούριο και καινοτόμο στον χώρο του Openstack αλλά και στην τεχνολογία του λογισμικού νέφους.

Release Name	Release Date
Austin	October 21, 2010
Bexar	February 3, 2011
Cactus	April 15, 2011
Diablo	September 22, 2011
Essex	April 5, 2012
Folsom	September 27, 2012
Grizzly	April 4, 2013
Havana	October 17, 2013
Icehouse	April 17, 2014
Juno	October 2014

5.3 Ημερομηνίες Εκδόσεων Openstack

5.2 Τα Openstack Projects

Η δομή του Openstack αποτελείται από Projects. Κάθε Project αποτελεί μια ξεχωριστή οντότητα με δικές της λειτουργίες και υπάρχει στο εσωτερικό του Openstack. Αυτή η διάρθρωση έχει ως αποτέλεσμα να απλοποιεί πάρα πολύ το περιβάλλον Openstack ως προς την ανάπτυξη, την αναβάθμιση και την συντήρηση του. Αξίζει να σημειωθεί ότι στις αρχικές εκδόσεις του Openstack , η αναβάθμιση του νέφους αποτελούσε μια πρώτης τάξεως πρόκληση για τους υπεύθυνους ενός συστήματος αλλά οι όποιες δυσκολίες δημιουργούνταν κάθε φορά , βελτιώθηκαν και ξεπεράστηκαν με τις μεταγενέστερες εκδόσεις.



5.4 Η δομή του Openstack

Όπως βλέπουμε και στην Εικόνα 6, παραπάνω , η δομή του Openstack είναι χωρισμένη σε 3 μέρη . Στο υπολογιστικό κομμάτι (**Compute**) ,στο κομμάτι του δικτύου (**Networking**) και στο κομμάτι που αφορά την αποθήκευση δεδομένων και πληροφοριών (**Storage**). Στο εσωτερικό κάθε ενός από τα 3 αυτά μέρη υπάρχουν υπηρεσίες που τα συνδέουν μεταξύ τους και στην συνέχεια η παράδοση του τελικού αποτελέσματος γίνεται μέσω ενός γραφικού περιβάλλοντος web που ονομάζεται **Dashboard** (αποτελεί Core Service το οποίο θα το δούμε αναλυτικά στην συνέχεια). Είναι αξιοσημείωτο το γεγονός ότι κάθε ένα από τα 3 παραπάνω μέρη είναι προσβάσιμο μέσω του δικού του API (Application Programming Interface).

Συνοψίζοντας, αυτή ήταν μια γενική όψη του οργανισμού Openstack. Επίσης είδαμε την λειτουργία και την δομή των πολύ βασικών συστατικών του. Στην συνέχεια θα αναλύσουμε τις υπηρεσίες που αποτελούν τον πυρήνα του Openstack και τα οποία χρησιμοποιήθηκαν από εμάς για την υλοποίηση του Lab-As-A-Service.

5.3 Ανάπτυξη ενός Περιβάλλοντος Νέφους με το Openstack

Καθώς προετοιμάζουμε την ανάπτυξη του δικού μας **Lab-As-A-Service** περιβάλλοντος είναι σημαντικό να καταλάβουμε ποιες είναι οι προαπαιτήσεις για την δημιουργία του. Υπάρχουν αρκετές υπηρεσίες με δικά τους χαρακτηριστικά στις οποίες στηριζόμαστε για την ανάπτυξη του περιβάλλοντος μας. Υπάρχουν υπηρεσίες οι οποίες εξαρτώνται από άλλες και απλά τις χρησιμοποιούμε όχι μόνο γιατί τις χρειαζόμαστε αλλά επίσης διότι χωρίς αυτές το σύστημα μας δεν θα είναι λειτουργικό. Οι περισσότερες υπηρεσίες είναι διαθέσιμες και έτοιμες προς χρήση μέσω των εκάστοτε εκδόσεων. Σε αρκετές περιπτώσεις όταν έχουμε να κάνουμε με ένα πολύπλοκο σύστημα , θα πρέπει να έχουμε την γνώση για το τι συμβαίνει στο υπόβαθρο αυτών των υπηρεσιών με σκοπό την καλύτερη διαχείριση και συντήρηση του συστήματος μας.

Αρχικά ,την βάση του συστήματος μας αποτελεί το λειτουργικό σύστημα που θα φιλοξενήσει το περιβάλλον νέφους. Επιλέξαμε την διανομή **Ubuntu Server** και πιο συγκεκριμένα την έκδοση 14.04. Βρήκαμε πολύ θετικό ότι η συγκεκριμένη έκδοση, σε αντίθεση με την προηγούμενη της , την 12.04 , προσφέρει τις απαραίτητες βιβλιοθήκες για την χρήση του Openstack έτοιμες κατά την

εγκατάσταση. Αξίζει να σημειωθεί επίσης ότι η έκδοση Ubuntu αποτελεί λειτουργικό σύστημα ελεύθερου κώδικα και ότι διαθέτει μεγάλη κοινότητα χρηστών που μπορεί να παρέχει πληροφορίες και βοήθεια τόσο για θέματα του λειτουργικού όσο και για το Openstack πιο συγκεκριμένα.

Στην συνέχεια είναι αναγκαία μια υπηρεσία για την διαχείριση Βάσεων Δεδομένων. Επιλέξαμε την MySQL[32] καθώς είναι η πλέον διαδεδομένη υπηρεσία για την διαχείριση Βάσεων Δεδομένων καθώς επίσης είναι και αυτή με την σειρά της λογισμικό ανοιχτού κώδικα. Παρόμοια λογισμικά αποτελούν το Firebird και η PostgreSQL.

Για την εγκατάσταση του database server χρησιμοποιήσαμε την εντολή:

```
# apt-get install mariadb-server python-mysqldb
```

Επίσης χρησιμοποιήσαμε και μια υπηρεσία για αλλαγή μηνυμάτων μεταξύ του Server και του Client. Η υπηρεσία αυτή, γνωστή και ως **Message Queueing[33]** αποτελεί έναν ασύγχρονο τρόπο επικοινωνίας μεταξύ Server και Client. Στην περίπτωση μας χρησιμοποιήσαμε την RabbitMQ . Είναι μια πολύ καλή λύση και χρησιμοποιείται από τους περισσότερους χρήστες που θέλουν να επωφεληθούν των πλεονεκτημάτων της συγκεκριμένης τεχνολογίας. Η RabbitMQ είναι και αυτή λογισμικό ανοιχτού κώδικα και χρησιμοποιείται από πολλά προϊόντα και υπηρεσίες. Τέλος, φυσικά εκτός από όλες αυτές της υπηρεσίες ανωτέρων επιπέδων , χρησιμοποιήσαμε τις κοινές υπηρεσίες στο τρίτο επίπεδο δικτύωσης για την δημιουργία του Lab-As-A-Service.

Για την εγκατάσταση του MQ χρησιμοποιήθηκε η εντολή:

```
# apt-get install rabbitmq-server
```

Ο message broker δημιουργεί έναν αρχικό λογαριασμό με όνομα και κωδικό χρήστη guest. Κρατήσαμε τον λογαριασμό guest απλά αλλάζοντας τον κωδικό του, αντικαθιστώντας το RABBIT_PASS της παρακάτω εντολής με τον κωδικό της επιλογής μας.

```
# rabbitmqctl change_password guest rabbitpass
```

Στο σημείο αυτό αξίζει να γίνει μια μικρή υποσημείωση σχετικά με το Openstack λογισμικό. Το Openstack χρησιμοποιεί **hypervisor[34] Πρώτου Τύπου** μοντέλο . Ο **Hypervisor** ή αλλιώς **Virtual Machine Monitor(VMM)** αποτελεί

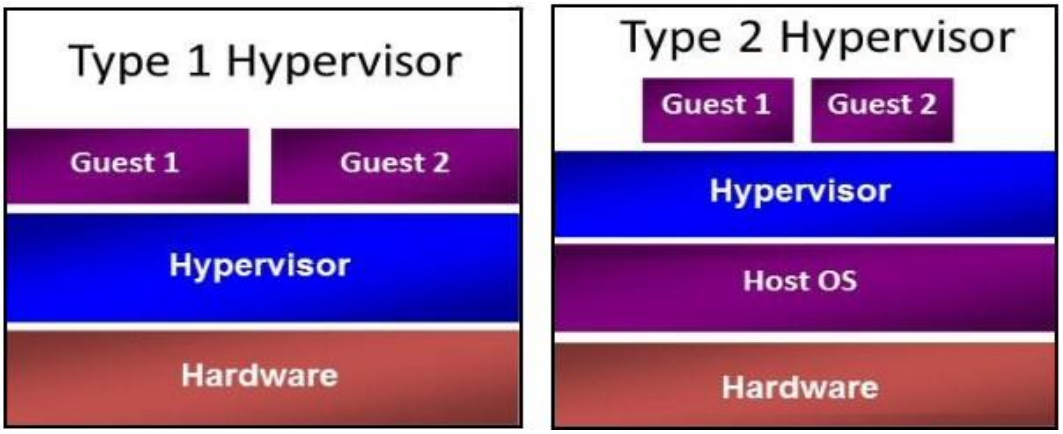
software, middleware ή hardware κομμάτι ενός υπολογιστικού συστήματος , το οποίο δημιουργεί και φιλοξενεί εικονικές μηχανές(Virtual Machines).

Ο **Πρώτος Τύπος** hypervisor, είναι γνωστός ως **Native** ή **Bare-Metal**.Αυτοί οι hypervisors είναι εγκατεστημένοι κατευθείαν πάνω στο Hardware και διαχειρίζονται τα εκάστοτε λειτουργικά συστήματα που φιλοξενούνται πάνω σε αυτούς και τα οποία αποτελούν διεργασίες.

Γνωστοί και διαδεδομένοι hypervisors Πρώτου Τύπου είναι οι:

- KVM (Kernel-Based Virtual Machine) για Linux.
- Red Hat Enterprise Virtualization (RHEV) το οποίο αποτελεί εφαρμογή της Red Hat πάνω στον KVM.
- Citrix XenServer.
- Hyper-V της Microsoft.
- vSphere-ESXi της VMware

Εκτός από τον Πρώτο Τύπο hypervisor ,υπάρχει και ο **Δεύτερος Τύπος** γνωστότερος με τον όρο **hosted hypervisor**.Σε αυτήν την περίπτωση ο hypervisor είναι εγκατεστημένος πάνω στο εκάστοτε λειτουργικό σύστημα (στην περίπτωση μας το Ubuntu Server 14.04) , όπως οποιοδήποτε άλλο λογισμικό. Οι πιο γνωστοί hosted hypervisors είναι το **VMware Workstation** και το **VirtualBox** της Oracle.



5.5 Πρώτος και Δεύτερος Τύπος Hypervisor

5.4 Οι Βασικές Υπηρεσίες του Openstack

Πριν ξεκινήσει να λειτουργεί το εγχείρημα μας στηριζόμενο στο Openstack Cloud , είναι αναγκαίες κάποιες από τις υπηρεσίες του που χωρίς αυτές κανένα Openstack Cloud δεν είναι λειτουργικό. Αυτές οι υπηρεσίες αποτελούν την ραχοκοκαλιά τόσο του δικού μας Lab-As-A-Service , όσο και κάθε άλλου περιβάλλοντος νέφους που βασίζεται στο Openstack.

Ένας ακόμα λόγος που καθιστά τις υπηρεσίες αυτές βασικές και αποτελούν τον πυρήνα του περιβάλλοντος νέφους μας, είναι το γεγονός ότι τα εργαλεία που παρέχουν είναι αυτά που χρειαζόμαστε έτσι ώστε να μπορούμε να διαχειριζόμαστε το περιβάλλον μας. Κάθε μία από αυτές τις υπηρεσίες είναι αναγκαία όχι μόνο για τον εαυτό της αλλά και από τις υπόλοιπες ακόμα και για την πιο βασική και απλή λειτουργία του περιβάλλοντος μας.

Παρακάτω θα δούμε αναλυτικά όλα αυτά τα συστατικά που χρησιμοποιήθηκαν τόσο από εμάς αλλά και από οποιονδήποτε άλλο χρήστη που θα χρησιμοποιεί ή θέλει να χρησιμοποιήσει το Openstack.

- **Keystone [35].** Αποτελεί την υπηρεσία της ασφάλειας του Νέφους μας τόσο για την **αυθεντικοποίηση** του χρήστη αλλά και για την **εξουσιοδότηση** των καθηκόντων του.
- **Glance [36].** Αυτή η υπηρεσία είναι αυτή που διαχειρίζεται και παραδίδει τις διάφορες εικόνες των λειτουργικών συστημάτων που υπάρχουν στην βάση δεδομένων του νέφους.
- **Nova[37].** Το Nova αποτελεί την επεξεργαστική υπηρεσία του περιβάλλοντος μας. Είναι η υπηρεσία που φιλοξενεί και διαχειρίζεται το υπολογιστικό μας σύστημα.
- **Neutron [38].** Είναι το κομμάτι του δικτύου. Σκοπός του Neutron είναι η διαχείριση των IP διευθύνσεων και των διαφόρων δικτύων στο εσωτερικό του περιβάλλοντος μας. Η διαχείριση αυτή γίνεται μέσω του API που διαθέτει.
- **Horizon (Dashboard) [39].** Αποτελεί την εικόνα του περιβάλλοντος μας , καθώς παρέχει γραφική απεικόνιση του συστήματος, τόσο στους διαχειριστές αλλά και στους χρήστες ώστε να μπορούν να το διαχειρίζονται και να το συντηρούν.

- **Cinder** [40]**.** Είναι η υπηρεσία ο οποία είναι υπεύθυνη για την διαχείριση των αποθηκευτικών μέσων που είναι διαθέσιμα προς χρήση.

5.4.1 Keystone

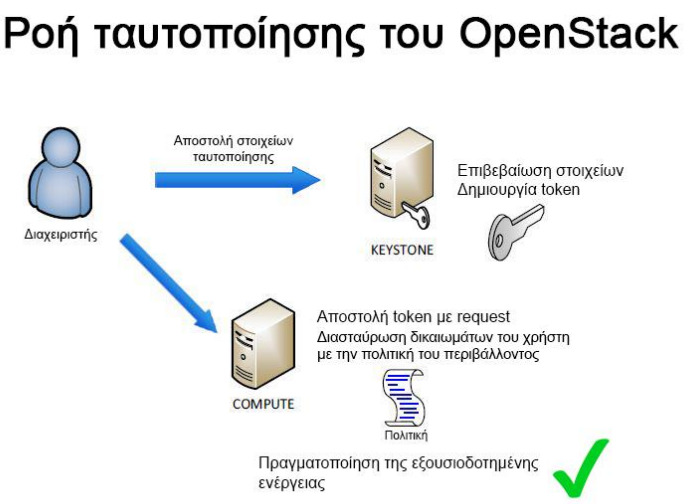
Η πρώτη από τις βασικές υπηρεσίες του Openstack έχει την κωδική ονομασία Keystone και αποτελεί την υπηρεσία ταυτοποίησης του χρήστη. Το Keystone θα μπορούσαμε να πούμε ότι αποτελεί την απάντηση στις ερωτήσεις “Ποιοι είμαστε;” και “Τι θέλουμε να κάνουμε ;” στο περιβάλλον μας.

Αρχικά , παρέχει Αυθεντικοποίηση το οποίο απαντάει στην πρώτη από τις δύο ερωτήσεις. Στην συνέχεια ελέγχει για την αρμοδιότητες του χρήστη και παρέχει την κατάλληλη εξουσιοδότηση βάσει πολιτικής, απαντώντας στο δεύτερο ερώτημα μας. Το Keystone περιλαμβάνει πολλά διαφορετικά κομμάτια του περιβάλλοντος μας. Εκτός από τους χρήστες , που είναι και το πιο προφανές, το Keystone μπορεί να παρέχει διαφορετική αυθεντικοποίηση και εξουσιοδότηση στις υπηρεσίες εντός του περιβάλλοντος αλλά και στα Endpoints.

Στο σημείο αυτό είναι καλό να αναφέρουμε τι είναι τα Endpoints. Τα Endpoints αποτελούν το σημείο επαφής ώστε οι διαφορετικές υπηρεσίες του Openstack να μπορούν να επικοινωνούν μεταξύ τους. Όπως μπορεί να γίνει κατανοητό όλα αυτά τα Endpoints είναι καταχωρημένα εντός του Keystone. Με αυτόν τον τρόπο γίνεται δυνατή η εύκολη ανάπτυξη του περιβάλλοντος μας .

Εκτός από την κλασσική αυθεντικοποίηση , το Keystone χρησιμοποιεί και Tokens με απώτερο σκοπό να αυθεντικοποιεί και να κρατά πληροφορίες των συνδέσεων που γίνονται στο νέφος. Αυτό είναι πολύ σημαντικό καθώς αποτελεί ένα παραπάνω επίπεδο ώστε να επιβεβαιώνει ποιος είναι αυτός που συνδέθηκε αλλά και να μην χρειάζεται να επιβεβαιώνει κάθε φορά ποιος είναι όταν κάνει χρήση κάποιας άλλης υπηρεσίας του περιβάλλοντος.

Ας δούμε ένα παράδειγμα που θα κάνει κατανοητό πως επιτυγχάνεται η αυθεντικοποίηση και η εξουσιοδότηση με το Keystone



5.6 Η ροή Ταυτοποίησης του Openstack

Ας υποθέσουμε ότι ένας χρήστης θέλει να κάνει σύνδεση στο openstack.it.teithe.gr για να πραγματοποιήσει κάποια λειτουργία εντός του openstack περιβάλλοντος του.

Αρχικά , για να γίνει εύκολα κατανοητό το παράδειγμα , θα το απλοποιήσουμε αναφέροντας μόνο το Keystone και το Compute περιβάλλον. Το πρώτο πράγμα που πρέπει να κάνει ο χρήστης είναι να συνδεθεί στο περιβάλλον και να στείλει τα στοιχεία σύνδεσης του (όνομα χρήστη και κωδικός χρήστη), με τα οποία θα γίνει η αυθεντικοποίηση του, στο περιβάλλον Keystone. Μετά την επιβεβαίωση , το Keystone δημιουργεί ένα token το οποίο επιστρέφεται πίσω στον χρήστη και το χρησιμοποιεί καθ' όλη την διάρκεια της σύνδεσης του στο περιβάλλον. Αυτό συμβαίνει έτσι ώστε όταν ζητά κάτι από το Compute περιβάλλον το token να στέλνεται μαζί με το Request . Έπειτα εμφανίζεται η εξουσιοδότηση. Σε αυτό το σημείο διασταυρώνονται τα δικαιώματα του χρήστη με την πολιτική του περιβάλλοντος Compute, που προέρχεται από το Keystone , όπου και διατηρείται. Έτσι αφού γίνει ο έλεγχος ότι ο χρήστης έχει το δικαίωμα να προβεί στην ζητούμενη ενέργεια , τότε και μόνο τότε η ενέργεια αυτή πραγματοποιείται.

Συνεχίζοντας με το Keystone , ένα ακόμα χαρακτηριστικό του είναι τα tenants(ενοικιαστές) . Τα tenant περιβάλλοντα τα συναντούμε σχεδόν αποκλειστικά σε συζητήσεις για νέφη και πιο συγκεκριμένα μιλάμε κυρίως για πολλαπλούς tenants. Οι Tenants είναι λογικές διαχωρισμένες μονάδες εντός του Openstack νέφους μας. Σαν γενικός όρος υφίσταται σε όλα τα περιβάλλοντα νέφους. Ο καλύτερος τρόπος για να περιγράψουμε τι είναι ο tenant είναι με το να

τον παρομοιάσουμε με ένα διαμέρισμα μιας πολυκατοικίας. Εμείς έχουμε ένα και μοναδικό περιβάλλον νέφους (πολυκατοικία) αλλά εντός αυτού υπάρχουν πολλοί tenants(διαμερίσματα) , οι οποίοι έχουν πρόσβαση στην δική τους περιοχή. Όλοι συνυπάρχουν στο ίδιο περιβάλλον αλλά δεν ξέρουν τις διαφορές που έχουν με τους άλλους tenants που υπάρχουν. Το μόνο που πρέπει να γνωρίζουν είναι ότι έχουν πρόσβαση στην δική τους εκάστοτε μονάδα και ότι μπορούν να κάνουν οποιαδήποτε ενέργεια εντός αυτής.

Η ιδιότητα των πολλαπλών tenants (multi-tenancy) εντός του keystone είναι σπουδαία καθώς αποτελεί έναν ασφαλή τρόπο για να κλειδώνει την αυθεντικοποίηση και την εξουσιοδότηση σε κάθε έναν tenant ξεχωριστά χωρίς να επηρεάζει όλους τους υπόλοιπους. Μπορούν να δοθούν διαφορετικά αντικείμενα προς χρήση σε κάθε έναν tenant , έτσι ώστε ο κάθε tenant με την σειρά του να δημιουργεί συγκεκριμένα αντικείμενα που μπορεί να τα διαμοιράσει σε άλλους tenants. Αυτός είναι ένας πολύ καλός τρόπος που μπορεί να λειτουργεί ένα περιβάλλον αλλά και ένας τρόπος για να το διαχειριζόμαστε εύκολα.

Επίσης ένας tenant μπορεί να δημιουργεί κάθε είδους διαφορετικό αντικείμενο, για παράδειγμα εικόνες λειτουργικών συστημάτων. Σε αυτό δεν θα εμπλέκεται κάνεις άλλος εκτός από τον συγκεκριμένο tenant χωρίς αυτό να αποτελεί πρόβλημα καθώς αυτό γίνεται εντός του περιβάλλοντος του tenant. Όλη η διεργασία συμβαίνει λόγω του ελέγχου ασφαλείας που μπορούμε να προσφέρουμε χρησιμοποιώντας το Keystone.

Για την εγκατάσταση και την υλοποίηση[41] του Keystone έγιναν οι παρακάτω ενέργειες:

Αρχικά, έπρεπε να δημιουργηθούν μια βάση δεδομένων και ένα administration token.

Για να δημιουργηθεί η βάση δεδομένων, πρώτα μέσω του mysql client συνδεθήκαμε ως root:

```
$ mysql -u root –p
```

Στην συνέχεια δημιουργήσαμε την βάση:

```
CREATE DATABASE keystone;
```

Δώσαμε την απαραίτητη πρόσβαση στην βάση:

```
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'195.251.123.247' \
                IDENTIFIED BY 'keystonepass';
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' \
                IDENTIFIED BY 'keystonepass';
```

Δημιουργήσαμε μία τυχαία δεκαψήφια, δεκαεξαδική τιμή για να την χρησιμοποιήσουμε σαν administration token κατά την διαδικασία των αρχικών ρυθμίσεων και παραμετροποιήσεων:

```
# openssl rand -hex 10
```

Αμέσως μετά κατεβάσαμε τα απαραίτητα πακέτα με την εντολή:

```
#apt-get install keystone python-keystoneclient
```

Παραμετροποιήσαμε το αρχείο /etc/keystone/keystone.conf ως εξής:

Στο [DEFAULT] τμήμα του αρχείου, ορίσαμε την αρχική τιμή του administration token που δημιουργήσαμε παραπάνω:

```
1    [DEFAULT]
2    . . .
3    admin_token = ADMIN_TOKEN
```

Στο [database] τμήμα του αρχείου, ρυθμίσαμε την πρόσβαση στην βάση δεδομένων:

```
1    [database]
2    . . .
3    connection=mysql://keystone:keystonepass@195.251.123.247/keystone
```

Στο [revoke] τμήμα του αρχείου, ρυθμίσαμε τον SQL revoking driver :

```
1    [revoke]
2    . . .
3    driver = keystone.contrib.revoke.backends.sql.Revoke
```

Προαιρετικά, για επιπλέον βοήθεια στην επίλυση προβλημάτων, ενεργοποιήσαμε το verbose στο [DEFAULT] τμήμα:

```
1    [DEFAULT]
2    . . .
3    verbose = True
```

Επαληθεύσαμε την δημιουργία της βάσης keystone μετρώντας το πλήθος της:

```
# su -s /bin/sh -c "keystone-manage db_sync" keystone
```

και στην συνέχεια επανεκκινήσαμε την υπηρεσία keystone:

```
# service keystone restart
```

Επόμενο βήμα μετά την εγκατάσταση του Identity Service, δημιουργήσαμε tenants (projects), χρήστες και ρόλους για το περιβάλλον μας. Εδώ να σημειώσουμε ότι χρησιμοποιούμε το προσωρινό administration token που δημιουργήσαμε προηγουμένως.

Αρχικά, διαμορφώσαμε τις προϋποθέσεις που έπρεπε να ισχύουν:

Διαμόρφωση του administration token:

```
$ export OS_SERVICE_TOKEN=ADMIN_TOKEN
```

Αντικαθιστώντας το ADMIN_TOKEN με την τιμή που δημιουργήσαμε, για παράδειγμα:

```
$ export OS_SERVICE_TOKEN=a591bc139df3183afa45
```

Διαμόρφωση του Endpoint:

```
$ export OS_SERVICE_ENDPOINT=http://195.251.123.247:35357/v2.0
```

Στην συνέχεια δημιουργήσαμε tenants, χρήστες και ρόλους.

Αρχικά, έπρεπε να δημιουργήσουμε ένα διαχειριστικό tenant, χρήστη και ρόλο που θα έχει όλα τα απαραίτητα δικαιώματα για να πράττει όλες τις λειτουργίες που απαιτούνται από έναν διαχειριστή του περιβάλλοντος.

Το admin tenant δημιουργήθηκε με την εντολή:

```
$ keystone tenant-create --name admin --description "Admin Tenant"
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |           Admin Tenant           |
|   enabled   |               True               |
|      id     | 6f4c1e4cbfef4d5a8a1345882fbca110 |
|     name    |              admin               |
+-------------+----------------------------------+
```

Το Openstack δημιουργεί τα IDs δυναμικά, για αυτόν τον λόγο θα φαίνονται διαφορετικές τιμές σε όλα τα παραδείγματα.

Στην συνέχεια δημιουργήσαμε τον `admin` user με το όρισμα:

```
$ keystone user-create --name admin --pass adminpass --email kkonstan@it.teithe.gr
```

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |    kkonstan@it.teithe.gr         |
| enabled  |               True               |
|    id    | ea8c352d253443118041c9c8b8416040 |
|   name   |              admin               |
| username |              admin               |
+----------+----------------------------------+
```

Επόμενο βήμα ήταν η δημιουργία του ρόλου admin:

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

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|    id    | bff3a6083b714fa29c9344bf8930d199 |
|   name   |              admin               |
+----------+----------------------------------+
```

Τέλος, θα πρέπει να προστεθεί ο admin ρόλος στον admin tenant και στον admin χρήστη:

```
$ keystone user-role-add --user admin --tenant admin --role admin
```

Αντίστοιχα με τον admin, δημιουργήσαμε ένα demo tenant και χρήστη για τυπικές λειτουργίες στο περιβάλλον μας.

Η δημιουργία έγινε με την εντολή:

```
$ keystone tenant-create --name demo --description "Demo Tenant"

+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |           Demo Tenant            |
|   enabled   |               True               |
|      id     | 4aa51bb942be4dd0ac0555d7591f80a6 |
|     name    |               demo               |
+-------------+----------------------------------+
```

Στην συνέχεια δημιουργήσαμε τον χρήστη demo μέσα στο demo tenant:

```
$ keystone user-create --name demo --tenant demo --pass demopass\
--email geosot@it.teithe.gr

+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |      geosot@it.teithe.gr         |
| enabled  |               True               |
|    id    | 7004dfa0dda84d63aef81cf7f100af01 |
|   name   |               demo               |
| tenantId | 4aa51bb942be4dd0ac0555d7591f80a6 |
| username |               demo               |
+----------+----------------------------------+
```

Αμέσως μετά την δημιουργία του admin και demo, δημιουργήσαμε το tenant της υπηρεσίας. Αυτό έγινε διότι το Openstack απαιτεί από τις υπηρεσίες του να έχουν tenant, χρήστη και ρόλο για να αλληλεπιδρούν μεταξύ τους. Συνήθως, από κάθε service απαιτείται η δημιουργία ενός ή παραπάνω μοναδικών χρηστών με ρόλο admin εντός του κάθε service tenant.

Η δημιουργία του tenant της υπηρεσίας πραγματοποιήθηκε με την εντολή:

```
$ keystone tenant-create --name service --description \
"Service Tenant"

+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |          Service Tenant          |
|   enabled   |               True               |
|      id     | 6b69202e1bf846a4ae50d65bc4789122 |
|     name    |             service              |
+-------------+----------------------------------+
```

Συνεχίζοντας την δημιουργία του Keystone, το επόμενο βήμα ήταν η δημιουργία των Endpoints για την υπηρεσία, για το API και την οντότητα της.

Το Keystone γενικά είναι αυτό που διαχειρίζεται έναν κατάλογο με τις υπόλοιπες υπηρεσίες εντός του περιβάλλοντος μας. Οι υπηρεσίες αυτές χρησιμοποιούν αυτόν τον κατάλογο για να εντοπίσουν τις υπόλοιπες με τις οποίες θα πρέπει να αλληλεπιδράσουν.

Η δημιουργία της οντότητας της υπηρεσίας έγινε με το όρισμα:

```
$ keystone service-create --name keystone --type identity \
--description "OpenStack Identity"

+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |        OpenStack Identity        |
|   enabled   |               True               |
|      id     | 4d3a687ab41ef47a2a190f2de613c12a |
|     name    |             keystone             |
|     type    |             identity             |
+-------------+----------------------------------+
```

Επίσης, η υπηρεσίας ταυτοποίησης Keystone είναι αυτή που διαχειρίζεται έναν κατάλογο με τα API Endpoints που σχετίζονται με τις υπηρεσίες του Openstack. Οι υπηρεσίες του Openstack, στην συνέχεια, συμβουλεύονται αυτόν τον κατάλογο για να δουν πως θα επικοινωνήσουν μεταξύ τους εντός του περιβάλλοντος.

Το Openstack παρέχει τρία διαφορετικά API Endpoints για κάθε υπηρεσία: admin, internal και public. Σε παραγωγικό επίπεδο, τα τρία αυτά Endpoints μπορεί να είναι διανεμημένα σε ξεχωριστά δίκτυα για λόγους ασφαλείας. Επίσης το Openstack υποστηρίζει πολλαπλά Regions για την επεκτασιμότητα του περιβάλλοντος. Για λόγους απλοποίησης, εμείς διαμορφώσαμε το περιβάλλον μας εντός ενός Region, του RegionOne.

Το API της υπηρεσίας ταυτοποίησης δημιουργήθηκε με την παρακάτω εντολή:

```
$ keystone endpoint-create \
--service-id $(keystone service-list | awk '/ identity / {print $2}') \
--publicurl http://195.251.123.247:5000/v2.0 \
--internalurl http:// 195.251.123.247:5000/v2.0 \
--adminurl http:// 195.251.123.247:35357/v2.0 \
--region RegionOne
```

```
+-------------+----------------------------------------+
|   Property  |              Value                     |
+-------------+----------------------------------------+
|   adminurl  |   http:// 195.251.123.247:35357/v2.0   |
|      id     |   11f9c625a3b94a3f8e66bf4e5de2679f     |
| internalurl |   http:// 195.251.123.247:5000/v2.0    |
|  publicurl  |   http:// 195.251.123.247:5000/v2.0    |
|    region   |              regionOne                 |
|  service_id |    4d3a687ab41ef47a2a190f2de613c12a    |
+-------------+----------------------------------------+
```

Τέλος, επαληθεύσαμε όλη την προηγούμενη διαδικασία για την σωστή λειτουργία του Keystone.

Αρχικά, αφαιρέσαμε τις προσωρινές μεταβλητές OS_SERVICE_TOKEN και OS_SERVICE_ENDPOINT με την εντολή:

```
$ unset OS_SERVICE_TOKEN OS_SERVICE_ENDPOINT
```

Ακολούθως, ως admin χρήστες, απαιτήσαμε από το σύστημα ένα token αυθεντικοποίησης ως εξής:

```
$ keystone --os-tenant-name admin --os-username admin \
--os-password adminpass --os-auth-url http://195.251.123.247:35357/v2.0 \ token-get
```

```
+-----------+----------------------------------+
|  Property |              Value               |
+-----------+----------------------------------+
|  expires  |       2015-12-14T11:22:12Z       |
|     id    | 5621afc2f1f8f7d48963eb5ccd864769 |
| tenant_id | 421aa4b4317ad5f90bc1a4542aa10401 |
|  user_id  | d5ff0b63a1a5b5475d09ca1098aab11b |
+-----------+----------------------------------+
```

Ως admin tenant και χρήστης, μπορούμε να δούμε την λίστα με τα tenants ώστε να επαληθεύσουμε ότι ο admin tenant και χρήστης μπορούν να εκτελέσουν admin-only CLI εντολές και να ελέγξουμε ότι η υπηρεσία ταυτοποίησης Keystone εμπεριέχει τα tenants που δημιουργήσαμε. Έτσι χρησιμοποιήσαμε την παρακάτω εντολή:

```
$ keystone --os-tenant-name admin --os-username admin --os-password \
adminpass  --os-auth-url http://195.251.123.247:35357/v2.0 tenant-list
```

```
+----------------------------------+----------+---------+
|                id                |   name   | enabled |
+----------------------------------+----------+---------+
| 421aa4b4317ad5f90bc1a4542aa10401 |  admin   |   True  |
| 4aa51bb942be4dd0ac0555d7591f80a6 |   demo   |   True  |
| 6b69202e1bf846a4ae50d65bc4789122 | service  |   True  |
+----------------------------------+----------+---------+
```

Επίσης, έχουμε την δυνατότητα ως admin χρήστης και tenant χρησιμοποιώντας την λίστα των χρηστών να επαληθεύουμε ότι το keystone εμπεριέχει τους χρήστες που δημιουργήσαμε, με την εντολή:

$ keystone --os-tenant-name admin --os-username admin --os-password \ adminpass --os-auth-url http://195.251.123.247:35357/v2.0 user-list

```
+----------------------------------+---------+---------+-----------------------+
|                id                |   name  | enabled |          email        |
+----------------------------------+---------+---------+-----------------------+
| d5ff0b63a1a5b5475d09ca1098aab11b |  admin  |   True  | kkonstan@it.teithe.gr |
| 7004dfa0dda84d63aef81cf7f100af01 |   demo  |   True  | geosot@it.teithe.gr   |
+----------------------------------+---------+---------+-----------------------+
```

Ακόμα, έχουμε την δυνατότητα ως admin tenant και χρήστης, με την λίστα των ρόλων να επαληθεύσουμε ότι το Keystone περιέχει τους ρόλους που δημιουργήσαμε με το όρισμα:

$ keystone --os-tenant-name admin --os-username admin --os-password \ adminpass --os-auth-url http://195.251.123.247:35357/v2.0 role-list

```
+----------------------------------+----------+
|                id                |   name   |
+----------------------------------+----------+
| 9fe2ff9ee4384b1894a90878d3e92bab | _member_ |
| bff3a6083b714fa29c9344bf8930d199 |  admin   |
+----------------------------------+----------+
```

Στην συνέχεια, ως demo tenant και χρήστης, πραγματοποιήθηκε αίτηση προς το σύστημα για ένα token αυθεντικοποίησης. Το όρισμα της αίτησης ήταν:

$ keystone --os-tenant-name demo --os-username demo --os-password \
demopass --os-auth-url http://195.251.123.247:35357/v2.0 token-get

```
+-----------+----------------------------------+
|  Property |              Value               |
+-----------+----------------------------------+
|  expires  |       2014-10-10T12:51:33Z       |
|     id    | 1b87ceae9e08411ba4a16e4dada04802 |
| tenant_id | 4aa51bb942be4dd0ac0555d7591f80a6 |
|  user_id  | 7004dfa0dda84d63aef81cf7f100af01 |
+-----------+----------------------------------+
```

Τέλος, ως demo tenant και χρήστης, προσπαθήσαμε να αιτηθούμε της λίστας των χρηστών και να επαληθεύσουμε ότι δεν μπορούμε να εκτελέσουμε admin-only CLI εντολές. Αυτό το πραγματοποιήσαμε με την εντολή:

```
$ keystone --os-tenant-name demo --os-username demo --os-password \
demopass --os-auth-url http://195.251.123.247:35357/v2.0 user-list

You are not authorized to perform the requested action, admin_required. (HTTP 403)
```

Όλες οι προηγούμενες ρυθμίσεις έγιναν μέσω του keystone client για να ρυθμίσουμε την υπηρεσία ταυτοποίησης του Openstack. Για να αυξήσουμε την αποτελεσματικότητα των εργασιών του client χρησιμοποιούμε απλά scripts που το Openstack υποστηρίζει. Αυτά τα scripts είναι γνωστά και ως αρχεία OpenRC. Συνήθως τα αρχεία αυτά εμπεριέχουν κοινές ρυθμίσεις για τους clients, υπάρχει όμως και η δυνατότητα υποστήριξης μοναδικών ρυθμίσεων.

Αρχικά δημιουργήσαμε τα δύο αρχεία, admin-openrc.sh και demo-openrc.sh για τους admin και demo αντίστοιχα. Στην συνέχεια, έγινε επεξεργασία του admin-openrc.sh και προστέθηκε το παρακάτω περιεχόμενο:

```
1  export OS_TENANT_NAME=admin
2  export OS_USERNAME=admin
3  export OS_PASSWORD=adminpass
4  export OS_AUTH_URL=http://195.251.123.247:35357/v2.0
```

Αντίστοιχα, για το demo-openrc.sh έγινε η προσθήκη του εξής περιεχομένου:

```
1  export OS_TENANT_NAME=demo
2  export OS_USERNAME=demo
3  export OS_PASSWORD=demopass
4  export OS_AUTH_URL=http:// 195.251.123.247:5000/v2.0
```

Επόμενο βήμα για να λειτουργούν οι client se συγκεκριμένο tenant και χρήστη, είναι να φορτωθούν τα scripts. Τα scripts αυτά θα ξεκινήσουν να λειτουργούν αφότου χρησιμοποιηθούν οι εντολές:

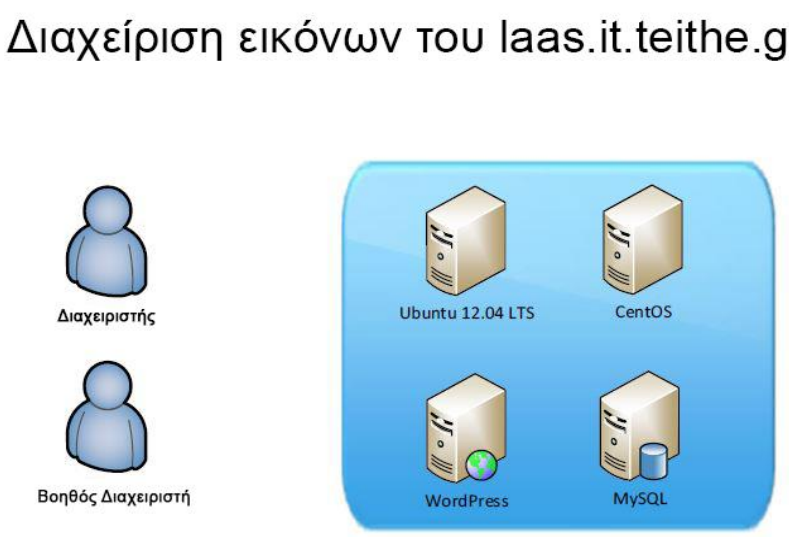
```
$ source admin-openrc.sh
```

5.4.2 Glance

Συνέχεια στην αναφορά των βασικών υπηρεσιών που απαρτίζουν το περιβάλλον μας , αποτελεί η υπηρεσίας διαχείρισης εικόνων με την κωδική ονομασία Glance. Σε αυτό το σημείο να κάνουμε ξεκάθαρο ότι όταν αναφερόμαστε στον όρο εικόνες δεν εννοούμε φωτογραφίες αποθηκευμένες στον δίσκο αλλά εικόνες εικονικών μηχανών[42]. Στο glance μας δίνεται η δυνατότητα να φιλοξενούμε εικόνες οι οποίες δεν είναι οι αρχικές που διαμοιράζονται από τους εκάστοτε οργανισμούς (π.χ.Ubuntu, CentOS, Windows) αλλά μπορούμε να φιλοξενούμε και τροποποιημένες εικόνες οι οποίες με την σειρά τους, φιλοξενούν εφαρμογές και προγράμματα της επιθυμίας μας.

Η υπηρεσία glance χρησιμοποιείται για να αποθηκεύει και να διαχειρίζεται τις εικόνες που δημιουργούνται .Τις εικόνες μπορούμε να τις διαχειριστούμε είτε συνολικά εντός του περιβάλλοντος είτε ξεχωριστά εντός κάποιου tenant. Αυτό μας δίνει την δυνατότητα να έχουμε εικόνες που μπορούν να χρησιμοποιηθούν απ' όλους εντός του νέφους αλλά και εικόνες που μπορούν να χρησιμοποιηθούν από συγκεκριμένους tenants. Επίσης μπορεί να δοθεί η κατάλληλη δικαιοδοσία σε συγκεκριμένους χρήστες έτσι ώστε να μπορούν να ανεβάζουν και αυτοί εικόνες στο περιβάλλον.

Η αποθήκευση των εικόνων μπορεί να γίνει σε διάφορα μέσα . Μπορούμε να χρησιμοποιήσουμε είτε τις υπηρεσίες Openstack Cinder και Swift είτε το τοπικό σύστημα αρχείων. Επίσης μας δίνεται η δυνατότητα της απομακρυσμένης αποθήκευσής τους για παράδειγμα στο AWS S3 (Amazon Simple Storage Service) ή οποιοδήποτε άλλο, παρόμοιο ,εξωτερικό περιβάλλον με σκοπό την εξοικονόμηση τοπικού αποθηκευτικού χώρου. Αυτός είναι ένας πολύ βολικός τρόπος όταν έχουμε μικρές και ελαφριές εικόνες που μπορούμε να τις αποθηκεύσουμε εκτός του προσωπικού μας περιβάλλοντος νέφους έχοντας πρόσβαση σε αυτές μέσω http.



5.7 Διαχείριση Glance Εικόνων εντός ενός Tenant

Ας υποθέσουμε ότι ό Διαχειριστής έχει το δικό του tenant περιβάλλον στο laas.it.teithe.gr και εντός αυτού του tenant υπάρχει μια εικόνα Ubuntu 12.04 την οποία χρησιμοποιεί για προσωπικές του ανάγκες και εργασίες και μια εικόνα CentOS που χρησιμοποιεί για τις ανάγκες εργασιών που απαιτούν CentOS σύστημα.

Επειδή ο Διαχειριστής χρειάζεται βοήθεια προσλαμβάνει τον Βοηθό Διαχειριστή ως διαχειριστή συστημάτων. Ο Βοηθός Διαχειριστή θέλει να φτιάξει τις προσωπικές του εικόνες βασισμένες πάνω σε αυτές του Διαχειριστή. Έτσι ό Βοηθός Διαχειριστή αποφασίζει να φτιάξει μια εικόνα WordPress με βάση την Ubuntu 12.04 εικόνα , όπως και ένα MySQL περιβάλλον βασισμένο στην εικόνα CentOS. Όλες αυτές οι εικόνες είναι διαθέσιμες από όλους εντός του tenant περιβάλλοντος. Με αυτόν τον τρόπο , εντός του tenant αυτού ο και οι δύο χρήστες έχουν το δικαίωμα να δημιουργούν και ανεβάζουν τις δικές τους προσωπικές εικόνες. Οι εικόνες αυτές δεν είναι αναγκαστικό να είναι διαθέσιμες εκτός του περιβάλλοντος και αυτός είναι ο απώτερος σκοπός όταν μιλάμε για εικόνες ανά tenant , οι οποίες είναι προσωπικές και δεν θέλουμε την διάδοσή τους παρά μόνο την ενθυλάκωσή τους εντός του tenant.

Για την εγκατάσταση και την ρύθμιση[43] του Glance χρειάζεται να γίνουν κάποιες ενέργειες που προαπαιτούνται ώστε αυτό να λειτουργήσει. Έτσι θα πρέπει να δημιουργήσουμε μια βάση δεδομένων, τα στοιχεία ταυτοποίησης της υπηρεσίας και το API Endpoint.

Αρχικά συνδεόμαστε στον client της βάσης δεδομένων ως root με την εντολή:

```
$ mysql -u root –p
```

Αμέσως επόμενο βήμα είναι η δημιουργία της βάσης glance:

```
CREATE DATABASE glance;
```

Μετά δίνουμε την κατάλληλη πρόσβαση στην βάση ως εξής:

```
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'195.251.123.247' \
        IDENTIFIED BY 'glancepass';
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' \
        IDENTIFIED BY 'glancepass';
```

Η επόμενη προαπαίτηση είναι η δημιουργία των στοιχείων ταυτοποίησης για το Glance. Για αυτόν τον σκοπό απαιτούνται κάποιες ενέργειες οι οποίες είναι:

1. Η δημιουργία του glance χρήστη με το όρισμα:

```
$ keystone user-create --name glance --pass glancepass
```

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |                                  |
| enabled  |               True               |
|    id    | f89cca5865dc42b18e2421fa5f5cce66 |
|   name   |              glance              |
| username |              glance              |
+----------+----------------------------------+
```

2. Η προσθήκη του ρόλου admin στον χρήστη glance:

```
$ keystone user-role-add --user glance --tenant service –role \ admin
```

3. Η δημιουργία της οντότητας της glance υπηρεσίας:

```
$ keystone service-create --name glance --type image \
  --description "OpenStack Image Service"
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |     OpenStack Image Service      |
|   enabled   |               True               |
|      id     | 23f409c4e79f4c9e9d23d809c50fbacf |
|     name    |              glance              |
|     type    |              image               |
+-------------+----------------------------------+
```

1. Η δημιουργία των Glance API Endpoints

```
$ keystone endpoint-create \
--service-id $(keystone service-list | awk '/ image / {print$2}')
--publicurl http://195.251.123.247:9292 \
--internalurl http:// 195.251.123.247:9292 \
--adminurl http:// 195.251.123.247:9292 \
--region regionOne
```

```
+-------------+-----------------------------------------+
|   Property  |                Value                    |
+-------------+-----------------------------------------+
|   adminurl  |   http:// 195.251.123.247:9292          |
|      id     | a2ee818c69cb475199a1ca108332eb35        |
| internalurl |   http:// 195.251.123.247:9292          |
|  publicurl  |   http:// 195.251.123.247:9292          |
|    region   |              regionOne                  |
|  service_id |    23f409c4e79f4c9e9d23d809c50fbacf     |
+-------------+-----------------------------------------+
```

Στην συνέχεια για την εγκατάσταση και την παραμετροποίηση του Glance, πρώτα κατεβάσαμε τα απαραίτητα πακέτα που ήταν απαραίτητα:

```
# apt-get install glance python-glanceclient
```

Μετά έγινε η επεξεργασία του αρχείου /etc/glance/glance-api.conf ως εξής:

a. Στο τμήμα [database], ρυθμίσαμε την πρόσβαση της βάσης δεδομένων:

```
1 [database]
2 . . .
3 connection=mysql://glance:glancepass@195.251.123.247/glance
```

b. Στο τμήμα [keystone_authtoken] και [paste_deploy], ρυθμίσαμε την πρόσβαση της υπηρεσίας ταυτοποίησης:

```
1       [keystone_authtoken]
2       . . .
3        auth_uri = http://195.251.123.247:5000/v2.0
4       identity_uri = http:// 195.251.123.247:35357
5       admin_tenant_name = service
6       admin_user = glance
7       admin_password = glancepass
8       [paste_deploy]
9       . . .
10      flavor = keystone
```

c. Στο τμήμα [glance_store] ρυθμίσαμε το τοπικό file system των image αρχείων:

```
1       [glance_store]
2       . . .
3       default_store = file
4       filesystem_store_datadir = /var/lib/glance/images/
```

Μετά από τις παραπάνω ρυθμίσεις, ρυθμίσαμε το αρχείο /etc/glance/glance-registry.conf ως εξής:

a. Στο τμήμα [database], ρυθμίστηκε η πρόσβαση της βάσης δεδομένων:

```
1 [database]
2 . . .
3 connection=mysql://glance:glancepass@195.251.123.247/glance
```

b. Στο τμήμα [keystone_authtoken] και [paste_deploy] έγιναν οι παρακάτω προσθήκες:

```
1   [keystone_authtoken]
2   . . .
3   auth_uri = http:// 195.251.123.247:5000/v2.0
4   identity_uri = http:// 195.251.123.247:35357
5   admin_tenant_name = service
6   admin_user = glance
7   admin_password = GLANCE_PASS
8   [paste_deploy]
9   . . .
10  flavor = keystone
```

Τέλος, για να ξεκινήσει η λειτουργία των υπηρεσιών του glance, τις επανεκκινήσαμε με τις εντολές:

```
# service glance-registry restart
# service glance-api restart
```

Για την επιβεβαίωση της σωστής λειτουργίας του Glance, χρησιμοποιήσαμε μια εικόνα Linux την CirrOS, η οποία βοηθάει σε αυτό.

Αρχικά δημιουργήσαμε έναν προσωρινό φάκελο για να αποθηκεύσουμε σε αυτόν την εικόνα:

```
$ mkdir /tmp/images
```

Στην συνέχεια, κατεβάσαμε την εικόνα CirrOS στην παραπάνω τοποθεσία που δημιουργήσαμε με την εντολή:

```
$ wget -P /tmp/images http://cdn.download.cirros- \
cloud.net/0.3.3/cirros-0.3.3-x86_64-disk.img
```

Επόμενο βήμα ήταν να αποκτήσουμε δικαιώματα admin για εντολές CLI μόνο:

```
$ source admin-openrc.sh
```

Στην συνέχεια ανεβάσαμε την εικόνα στο Glance, χρησιμοποιώντας το όρισμα:

```
$ glance image-create --name "cirros-0.3.3-x86_64" –file \
/tmp/images/cirros-0.3.3-x86_64-disk.img \
  --disk-format qcow2 --container-format bare --is-public True --progress
```

```
[==============================>] 100%
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 133eae9fb1c98f45894a4e60d8736619     |
| container_format | bare                                 |
| created_at       | 2014-10-10T13:14:42                  |
| deleted          | False                                |
| deleted_at       | None                                 |
| disk_format      | qcow2                                |
| id               | acafc7c0-40aa-4026-9673-b879898e1fc2 |
| is_public        | True                                 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | cirros-0.3.3-x86_64                  |
| owner            | ea8c352d253443118041c9c8b8416040     |
| protected        | False                                |
| size             | 13200896                             |
| status           | active                               |
| updated_at       | 2014-11-12T13:14:43                  |
| virtual_size     | None                                 |
+------------------+--------------------------------------+
```

Τέλος για την επαλήθευση της παραπάνω εντολής, μπορούμε να αιτήσουμε την λίστα με τις εικόνες που είναι καταχωρημένες στην υπηρεσία. Αυτό γίνεται με την εντολή:

```
$ glance image-list
```

Για την εγκατάσταση Ubuntu[44] εικόνας στο glance, ακολουθήθηκε η εξής διαδικασία:

Αρχικά, έγινε λήψη της εικόνας από το αρχείο της διανομής που βρίσκεται στον ιστότοπο http://archive.ubuntu.com/ubuntu/dists/trusty/main/installer-amd64/current/images/netboot/mini.iso .

Στην συνέχεια δημιουργούμε έναν εικονικό αποθηκευτικό χώρο με την εντολή:

```
# qemu-img create -f qcow2 /tmp/trusty.qcow2 10G
```

και ξεκινάμε την εγκατάσταση σε αυτόν τον αποθηκευτικό χώρο με την εντολή:

```
# virt-install --virt-type kvm --name trusty --ram 1024 \
        --cdrom=/data/isos/trusty-64-mini.iso \
        --disk /tmp/trusty.qcow2,format=qcow2 \
        --network network=default \
        --graphics vnc,listen=0.0.0.0 --noautoconsole \
        --os-type=linux --os-variant=ubuntutrusty
```

Με έναν VNC viewer πληκτρολογώντας την IP 0.0.0.0 θα μεταβούμε στο γραφικό περιβάλλον εγκατάστασης του λειτουργικού συστήματος. Μετά το πέρας και την ολοκλήρωση της εγκατάστασης χρησιμοποιούμε την παρακάτω εντολή για να βγάλουμε την εικόνα από τον δίσκο:

```
# virsh dumpxml trusty

<domain type='kvm'>
        <name>trusty</name>
...
 <disk type='block' device='cdrom'>
    <driver name='qemu' type='raw'/>
    <target dev='hdc' bus='ide'/>
    <readonly/>
    <address type='drive' controller='0' bus='1' target='0' unit='0'/>
        </disk>
...
</domain>
```

Επόμενο βήμα είναι να δώσουμε τις παρακάτω εντολές ως root, για να επανεκκινήσουμε την μηχανή με την ένδειξη paused και στην συνέχεια να βγάλουμε τον δίσκο και να συνεχίσουμε την διαδικασία

```
# virsh start trusty --paused
# virsh attach-disk --type cdrom --mode readonly trusty "" hdc
# virsh resume trusty
```

Αμέσως μετά εισερχόμαστε στην καινούρια εικόνα που μόλις δημιουργήσαμε. Κατά την πρώτη εκκίνηση, εισερχόμαστε ως root user και κάνουμε λήψη των πακέτων cloud-init:

```
# apt-get install cloud-init
```

Κατά την δημιουργία Ubuntu εικόνων το cloud-init πρέπει να ρυθμιστεί κατάλληλα με την πηγή των μεταδεδομένων. Αυτό γίνεται με την εντολή:

```
# dpkg-reconfigure cloud-init
```

Στην συνέχεια απενεργοποιούμε το instance ως root με την εντολή:

```
# /sbin/shutdown -h now
```

και αφαιρούμε όλες τις εγγραφές MAC διευθύνσεων:

```
# virt-sysprep -d trusty
```

Στο σημείο αυτό είμαστε έτοιμοι για να ανεβάσουμε την εικόνα μας στο Glance.

```
$ glance image-create --name "Ubuntu x64" --file  /tmp/trusty.qcow2 \
-disk-format qcow2 --container-format bare --is-public True
```

Για την εγκατάσταση[45] μια εικόνας Windows OS ακολουθήθηκαν τα παρακάτω βήματα:

Αρχικά κατεβάσαμε μια εικόνα Windows από τον ιστότοπο http://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2012 και την εικόνα με τους VirtIO οδηγούς από το http://alt.fedoraproject.org/pub/alt/virtio-win/latest/images/ .

Αμέσως μετά, δημιουργήσαμε μια εικόνα qcow2:

```
$ qemu-img create -f qcow2 ws2012.qcow2 15G
```

Και ξεκινήσαμε την εγκατάσταση με την εντολή virt-install:

```
# virt-install --connect qemu:///system \
--name ws2012 --ram 2048 --vcpus 2 \
--network network=default,model=virtio \
--disk path=ws2012.qcow2,format=qcow2,device=disk,bus=virtio \
--cdrom /path/to/en_windows_server_2012_x64_dvd.iso \
--disk path=/path/to/virtio-win-0.1-XX.iso,device=cdrom \
--vnc --os-type windows --os-variant win2k8
```

Αφού συνδεθούμε στην μηχανή ξεκινάμε την εγκατάσταση πρέπει να ενεργοποιήσουμε τους οδηγούς VirtIO. Ο δίσκος ως προεπιλογή δεν αναγνωρίζεται από τον Windows Installer. Όταν ερωτηθούμε για τον installation target, επιλέγουμε το κουμπί **Load Driver** και επιλέγουμε από το File system τον φάκελο `E:\WIN8\AMD64`. Ο Windows Installer προβάλλει μια λίστα από οδηγούς προς εγκατάσταση. Εμείς επιλέγουμε τους VirtIO SCSI και network οδηγούς και συνεχίζουμε την εγκατάσταση. Στην συνέχεια συνδεόμαστε ως administrator και ολοκληρώνουμε την εγκατάσταση δίνοντας την παρακάτω εντολή σε ένα cmd window:

```
C:\pnputil -i -a E:\WIN8\AMD64\*.INF
```

Για να επιτρέπουμε στο Cloudbase-init να τρέχει scripts κατά το boot ενός instance, θέτουμε την πολιτική εκτέλεσης του Powershell σε unrestricted:

```
C:\powershell
C:\Set-ExecutionPolicy Unrestricted
```

Επόμενο βήμα είναι η λήψη και η εγκατάσταση του Cloudbase-Init:

```
C:\Invoke-WebRequest -UseBasicParsing \
http://www.cloudbase.it/downloads/CloudbaseInitSetup_Beta_x64.msi \
 -OutFile cloudbaseinit.msi C:\.\cloudbaseinit.msi
```

Στο παράθυρο ρυθμίσεων, αλλάζουμε τα παρακάτω πεδία:

- Username: Administrator
- Network Adapter to configure: Red Hat VirtIO Ethernet Adapter
- Serial port for logging: COM1

Μετά την ολοκλήρωση της εγκατάστασης, στο παράθυρο του **Complete the Cloudbase-Init Setup Wizard,** επιλέγουμε τα checkboxes **Run Sysprep** και **Shutdown** και τερματίζουμε την διαδικασία.

Τέλος η εικόνα μας είναι έτοιμη για να ανέβει στο Glance:

```
$ glance image-create --name WS2012 --disk-format qcow2 \
 --container-format bare --is-public true \
 --file ws2012.qcow2
```

5.4.3 Nova

Πριν ξεκινήσουμε να μιλάμε για την υπηρεσία Nova που αποτελεί την επεξεργαστική δύναμη του συστήματος μας θα θέλαμε να κάνουμε ένα μικρό σχόλιο σχετικά με το περιβάλλον Nova και το openstack έτσι ώστε να γίνει ξεκάθαρη η υπόσταση του καθενός αλλά και οι ρόλοι που έχουν . Το openstack αποτελεί ένα οικοσύστημα νέφους το οποίο χρειάζεται έναν hypervisor για την υπολογιστική του πλατφόρμα (Nova). Η υπηρεσία Nova δεν είναι hypervisor αλλά αποτελεί κατά κάποιο τρόπο τον πίνακα ελέγχου διαχείρισης ενός hypervisor που βρίσκεται στη βάση.

Η υπηρεσία της επεξεργαστικής δύναμης του συστήματος νέφους μας αποτελεί την πλατφόρμα πάνω στην οποία θα φιλοξενούνται και θα λειτουργούν οι εικονικές μηχανές μας. Σαν ορολογία θα αναφερόμασταν σε αυτές τις μηχανές ως **“booting instances”** . Αυτά τα instances θα προέρχονταν από εικόνες που υπάρχουν στο Glance. Όπως είδαμε προηγουμένως στο παράδειγμα του Glance(Εικόνα 9) μπορούμε λόγου χάρη, βασιζόμενοι στις εικόνες του Ubuntu 14.04 ή του WordPress να δημιουργήσουμε instances των εικόνων αυτών. Στην βάση των instances αυτών θα υπάρχει ένας hypervisor που ο ρόλος του είναι να φιλοξενεί αυτές τις εικόνες. Το Openstack έχει την δυνατότητα να φιλοξενεί ταυτόχρονα πολλαπλούς hypervisors, δίνοντας μας το προνόμιο να χρησιμοποιούμε διαφορετικούς αν το επιθυμούμε. Για παράδειγμα μπορούμε να τρέχουμε hypervisors είτε ανοιχτού κώδικα είτε όχι όπως είναι οι KVM, XEN, V-Sphere, Hyper-V τους οποίους υποστηρίζει το Openstack.

Όσων αφορούν τους hypervisors και το Nova , ένας κατά κάποιον τρόπος περιορισμός αποτελεί το γεγονός ότι χρειαζόμαστε διαφορετικό Nova instance για κάθε ξεχωριστό hypervisor εντός του περιβάλλοντος μας. Αυτό πρακτικά δεν είναι κακό , απλώς είναι σημαντικό θα πρέπει να γνωρίζουμε ότι κάθε nova instance έχει διαφορετικές ρυθμίσεις και παραμέτρους λόγω του διαφορετικού οδηγού(driver) που διαθέτει ο εκάστοτε hypervisor.

Τέλος θα θέλαμε να αναφέρουμε τρεις ορολογίες για το Openstack και πως αυτές μπορούν να μας δώσουν μια καλύτερη εικόνα κατανόησης του περιβάλλοντος μας. Οι όροι αυτοί είναι οι **Regions**, **Aggregates** και **Availability Zones**.

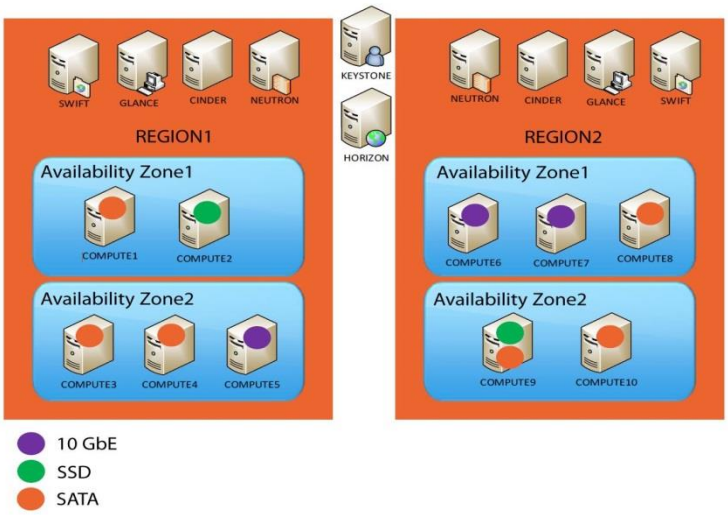
Ως **Regions** μπορούμε να ορίσουμε λογικές περιοχές που αποτελούνται από υπηρεσίες του Openstack. Δεν υπάρχει φυσική υπόσταση του Region , αλλά ως λογική ερμηνεία ορίζεται ως μια περιοχή που περιέχει όλες εκείνες τις υπηρεσίες του Openstack που χρειάζεται για να λειτουργήσει το συγκεκριμένο Region.

Ως **Aggregates** μπορούμε να ορίσουμε ομάδες ή ένα σύνολο από Endpoints τα οποία βασίζονται σε ένα φυσικό χαρακτηριστικό. Το χαρακτηριστικό με το οποίο γίνεται η ομαδοποίηση μπορεί να είναι οποιοδήποτε. Για παράδειγμα η χρήση SSD σκληρών δίσκων , η χρήση γρήγορων δικτύων και δικτυακών

συσκευών(10 GbE NICs) ή ακόμα αν χρησιμοποιούμε επεξεργαστές 12 και 6 πυρήνων , μπορούμε να ομαδοποιήσουμε ανάλογα με τον αριθμό αυτό σε διαφορετικό Aggregate.

Τελευταίο όρο αποτελεί το **Availability Zone**. Τυπικά θα μπορούμε να αναφερθούμε στο Availability Zone ως μια έννοια η οποία περιγράφει ένα φυσικό Region όπου σε αυτό συνυπάρχουν πολλά Openstack Endpoints.

Ακολουθεί μια απεικόνιση των τριών παραπάνω εννοιών αλλά και μια περιγραφή για να μας δώσει να καταλάβουμε αυτές τις έννοιες καλύτερα.



5.8 Regions, Aggregates και Availability Zones

Αρχικά έχουμε το περιβάλλον Keystone το οποίο είναι αρμόδιο για την ταυτοποίηση και την εξουσιοδότηση εντός του συστήματος μας. Στην συνέχεια έχουμε το περιβάλλον Horizon , για το οποίο θα μιλήσουμε στην συνέχεια και αποτελεί την διεπαφή διαχείρισης του συστήματος. Εντός του Keystone δημιουργούμε τις δύο λογικές περιοχές με ονόματα REGION1 και REGION2 αντίστοιχα. Στην συνέχεια, στο εσωτερικό του εκάστοτε Region, δημιουργούμε Endpoints για να παρέχουμε τις υπηρεσίες που επιθυμούμε σε καθένα από αυτά (στο συγκεκριμένο παράδειγμα παρέχουμε τις Swift, Glace, Cinder, Neutron και στα δύο Regions αλλά οι υπηρεσίες μπορεί να διαφέρουν). Επίσης , στο εσωτερικό κάθε Region υπάρχουν Availability Zones (στο παράδειγμα μας Availability Zone1 και Availability Zone2 για κάθε Region) , όπου εντός κάθε Availability Zone έχουν αναπτυχθεί και φιλοξενεί με την σειρά του Nova

Endpoints.Με αυτόν τον τρόπο όλες οι Nova υπηρεσίες μας τρέχουν σε συγκεκριμένες περιοχές που ορίσαμε εμείς.

Με αυτόν τον τρόπο μπορούμε να καταλάβουμε καλύτερα την λογική δομή του περιβάλλοντος μας όπως επίσης να έχουμε και καλύτερη εικόνα κατά την κατάστρωση της φυσικής δομής μέσω της γραφικής αναπαράστασης. Όμως δεν θα πρέπει να ξεχνάμε ότι υπάρχουν και τα Aggregates. Δεν υπάρχει όριο στον αριθμό που επιθυμούμε να έχουμε. Έτσι μπορούμε να δημιουργήσουμε όσα εμείς επιθυμούμε, βασισμένα πάντα σε φυσικά χαρακτηριστικά. Στο παράδειγμα μας φαίνεται με μωβ χρώμα το Aggregate των Nova Endpoints που χρησιμοποιούν κάρτες δικτύου των 10 Gigabit, ενώ τα υπόλοιπα Endpoints μπορεί να έχουν του 1 Gigabit. Εν συνεχεία, με πράσινη ένδειξη το Aggregate που δηλώνει την χρήση SSD σκληρών δίσκων και με πορτοκαλί χρώμα το Aggregate όσων Nova Endpoints χρησιμοποιούν την παραδοσιακή SATA τεχνολογία. Τέλος θα θέλαμε να επισημάνουμε ότι μερικά Endpoints ανήκουν σε περισσότερα από ένα Aggregates και αυτό συμβαίνει διότι κάθε Nova περιβάλλον έχει παραπάνω από ένα φυσικά χαρακτηριστικά.

Για την εγκατάσταση και την ρύθμιση[46] του Nova χρειάζεται να γίνουν κάποιες ενέργειες που προαπαιτούνται ώστε αυτό να λειτουργήσει, όπως ακριβώς συνέβη και με το glance και το keystone. Έτσι θα πρέπει να δημιουργήσουμε μια βάση δεδομένων, τα στοιχεία ταυτοποίησης της υπηρεσίας και τα API Endpoints. Για το Nova απαιτείται η εγκατάσταση τόσο του κόμβου ελέγχου(Controller Node), όσο και ενός υπολογιστικού κόμβου(Compute Node). Στην δική μας περίπτωση αυτοί οι δύο κόμβοι φιλοξενούνται στην ίδια μηχανή, οπότε οι ρυθμίσεις έγιναν με βάση αυτήν.

Αρχικά συνδεόμαστε στον client της βάσης δεδομένων ως root με την εντολή:

```
$ mysql -u root –p
```

Αμέσως επόμενο βήμα είναι η δημιουργία της βάσης nova:

```
CREATE DATABASE nova;
```

Μετά δίνουμε την κατάλληλη πρόσβαση στην βάση ως εξής:

```
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'195.251.123.247' \
```

```
        IDENTIFIED BY 'novapass';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' \
        IDENTIFIED BY 'novapass';
```

Η επόμενη προαπαίτηση είναι η δημιουργία των στοιχείων ταυτοποίησης για το Glance. Για αυτόν τον σκοπό απαιτούνται κάποιες ενέργειες οι οποίες είναι:

1. Η δημιουργία του glance χρήστη με το όρισμα:

```
$ keystone user-create --name nova --pass novapass
```

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |                                  |
| enabled  |               True               |
|    id    | 387dd4f7e46d4f72965ee99c76ae748c |
|   name   |               nova               |
| username |               nova               |
+----------+----------------------------------+
```

2. Η προσθήκη του ρόλου admin στον χρήστη nova:

```
$ keystone user-role-add --user nova --tenant service –role \ admin
```

3. Η δημιουργία της οντότητας της glance υπηρεσίας:

```
$ keystone service-create --name nova --type image \
--description "OpenStack Compute"
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |        OpenStack Compute         |
|   enabled   |               True               |
|      id     | 6c7854f52ce84db795557ebc0373f6b9 |
|     name    |               nova               |
|     type    |             compute              |
+-------------+----------------------------------+
```

4. Η δημιουργία των Glance API Endpoints

```
$ keystone endpoint-create \
  --service-id $(keystone service-list | awk '/ compute / \    {print$2}')
  --publicurl http://195.251.123.247:8774/v2/%\(tenant_id\)s \
  --internalurl http:// 195.251.123.247:8774/v2/%\(tenant_id\)s\
  --adminurl http:// 195.251.123.247:8774/v2/%\(tenant_id\)s \
  --region regionOne
```

Στην συνέχεια για την εγκατάσταση και την παραμετροποίηση του Nova, πρώτα κατεβάσαμε τα πακέτα που ήταν απαραίτητα:

```
# apt-get install nova-api nova-cert nova-conductor \
nova-consoleauth nova-novncproxy nova-scheduler python-novaclient
```

Μετά από τις παραπάνω ρυθμίσεις, ρυθμίσαμε το αρχείο /etc/nova/nova.conf ως εξής:

a. Στο τμήμα [database], ρυθμίστηκε η πρόσβαση στην βάση δεδομένων:

```
1  [database]
2  . . .
3  connection=mysql://nova:novapass@195.251.123.247/nova
```

b. Στο τμήμα [keystone_authtoken] και [DEFAULT] έγιναν οι παρακάτω προσθήκες:

```
1   [keystone_authtoken]
2   . . .
3   auth_uri = http:// 195.251.123.247:5000/v2.0
4   identity_uri = http:// 195.251.123.247:35357
5   admin_tenant_name = service
6   admin_user = nova
7   admin_password = novapass
8   [DEFAULT]
9   . . .
10  auth_strategy = keystone
```

c. Επίσης, στο [DEFAULT] τμήμα, έγιναν οι απαραίτητες ρυθμίσεις για την πρόσβαση της RabbitMQ:

```
1    [DEFAULT]
2    . . .
3    rpc_backend = rabbit
4    rabbit_host = 195.251.123.247
5    rabbit_password = rabbitpass
```

d. Στο [DEFAULT] τμήμα, ρυθμίστηκε η επιλογή my_ip ώστε να είναι η IP διεύθυνση του management interface του κόμβου ελέγχου, που στην προκειμένη είναι ο ίδιος μιας και η υλοποίηση μας γίνεται με την χρήση ενός και μοναδικού κόμβου.

```
1    [DEFAULT]
2    . . .
3    my_ip = 195.251.123.247
```

e. Στο [DEFAULT] τμήμα, προστέθηκαν οι ρυθμίσεις για τον VNC proxy έτσι ώστε να χρησιμοποιεί και αυτός την IP διεύθυνση του management interface του κόμβου ελέγχου:

```
1    [DEFAULT]
2    . . .
3    vncserver_listen = 195.251.123.247
4    vncserver_proxyclient_address = 195.251.123.247
```

f. Συνεχίζοντας, στο [glance] τμήμα, ρυθμίσαμε την τοποθεσία στην οποία βρίσκεται η υπηρεσία των εικόνων(Glance):

```
1    [glance]
2    . . .
3    Host= 195.251.123.247
```

Τέλος, για να ξεκινήσει η λειτουργία των υπηρεσιών του Nova, τις επανεκκινήσαμε με τις εντολές:

```
# service nova-api restart
# service nova-cert restart
# service nova-consoleauth restart
# service nova-scheduler restart
# service nova-conductor restart
# service nova-novncproxy restart
```

Επόμενο βήμα ήταν η ρύθμιση του υπολογιστικού κόμβου (Compute Node). Για την εγκατάσταση του, σαν πρώτο βήμα ήταν να κατεβάσουμε τα πακέτα εκείνα που χρειάζονταν για την ρύθμιση των συστατικών του hypervisor

QEMU μαζί με extensions του KVM στον υπολογιστικό κόμβο, καθώς ο KVM υποστηρίζει hardware acceleration για τις εικονικές μηχανές. Τα πακέτα ελήφθησαν με την εντολή:

```
# apt-get install nova-compute sysfsutils
```

Μετά από λήψη των πακέτων, ρυθμίσαμε το αρχείο /etc/nova/nova.conf ως εξής:

a. Στο [DEFAULT] τμήμα, έγιναν οι απαραίτητες ρυθμίσεις για την πρόσβαση της RabbitMQ:

```
1   [DEFAULT]
2   . . .
3   rpc_backend = rabbit
4   rabbit_host = 195.251.123.247
5   rabbit_password = rabbitpass
```

b. Στο [DEFAULT] και [keystone_authtoken] τμήμα, έγιναν οι ρυθμίσεις που αφορούσαν την υπηρεσία ταυτοποίησης (Keystone):

```
1   [DEFAULT]
2   . . .
3   auth_strategy = keystone
4   [keystone_authtoken]
5   . . .
6   auth_uri = http://195.251.123.247:5000/v2.0
7   identity_uri = http:// 195.251.123.247:35357
8   admin_tenant_name = service
9   admin_user = nova
10  admin_password = novapass
```

c. Στο [DEFAULT] τμήμα, προστέθηκε επίσης η επιλογή my_ip:

```
1   [DEFAULT]
2   . . .
3   my_ip = 195.251.123.247
```

d. Η τελευταία ρύθμιση στο [DEFAULT] τμήμα, αφορούσε την απομακρυσμένη πρόσβαση μέσω κονσόλας:

```
1 [DEFAULT]
2 . . .
3 vnc_enabled = True
4 vncserver_listen = 0.0.0.0
5 vncserver_proxyclient_address = 195.251.123.247
6 novncproxy_base_url=http://195.251.123.247:6080/vnc_auto.html
```

e. Στο [glance] τμήμα ρυθμίσαμε την τοποθεσία που υπάρχει η υπηρεσία διαχείρισης των εικόνων (Glance):

```
1 [DEFAULT]
2 . . .
3 host = 195.251.123.247
```

Για τον τερματισμό της εγκατάστασης, διασταυρώσαμε ότι ο υπολογιστικός κόμβος υποστηρίζει hardware acceleration για τις εικονικές μηχανές με την παρακάτω εντολή:

```
$ egrep -c '(vmx|svm)' /proc/cpuinfo
```

Αν η εντολή επιστρέψει τιμή μεγαλύτερη ή ίση με την μονάδα τότε ο κόμβος υποστηρίζει hardware acceleration και δεν χρειάζεται περαιτέρω ρύθμιση. Αν όμως η εντολή επιστρέψει τιμή ίση με το μηδέν ο κόμβος δεν υποστηρίζει hardware acceleration και πρέπει να τροποποιήσουμε το libvirt τμήμα για να χρησιμοποιήσει τον QEMU αντί του KVM. Αυτό γίνεται ως εξής:

Αρχικά, τροποποιούμε το τμήμα [libvirt] του αρχείου /etc/nova/nova-compute.conf.

```
1 [libvirt]
2 . . .
3 virt_type = qemu
```

Στην συνέχεια επανεκκινούμε την υπηρεσία Nova:

```
# service nova-compute restart
```

Για την επαλήθευση της διαδικασίας δώσαμε την εντολή για να δούμε την λίστα με τις nova υπηρεσίες που είναι ενεργές:

```
$ nova service-list
```

```
+----+------------------+------------+----------+---------+-------+
| Id | Binary           | Host       | Zone     | Status  | State |
+----+------------------+------------+----------+---------+-------+
| 1  | nova-conductor   | controller | internal | enabled | up    |
| 2  | nova-consoleauth | controller | internal | enabled | up    |
| 3  | nova-scheduler   | controller | internal | enabled | up    |
| 4  | nova-cert        | controller | internal | enabled | up    |
| 5  | nova-compute     | compute1   | nova     | enabled | up    |
+----+------------------+------------+----------+---------+-------+
```

Επίσης, μπορούμε να δούμε την λίστα των εικόνων του Glance για να επαληθεύσουμε την σύνδεση μεταξύ του Nova και του Glance:

```
$ nova image-list
```

```
+--------------------------------------+----------------------+--------+--------+
| ID                                   | Name                 | Status | Server |
+--------------------------------------+----------------------+--------+--------+
| acafc7c0-40aa-4026-9673-b879898e1fc2 | cirros-0.3.3-x86_64 | ACTIVE |        |
+--------------------------------------+----------------------+--------+--------+
```

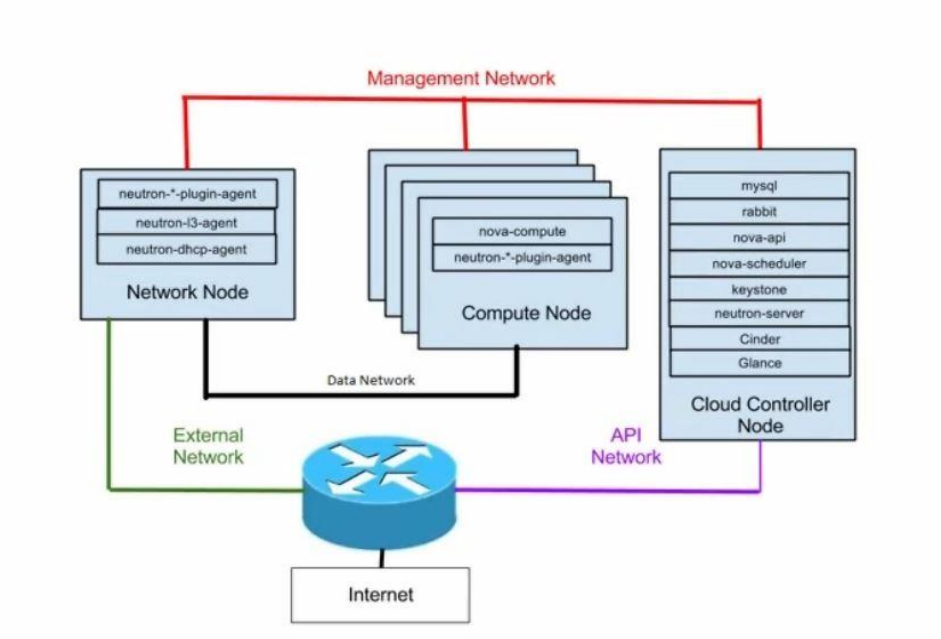
5.4.4 Neutron

Συνεχίζοντας με τις βασικές υπηρεσίες που χρησιμοποιήσαμε , θα αναφερθούμε σε αυτήν που είναι υπεύθυνη για το δικτυακό κομμάτι του περιβάλλοντος , είτε εσωτερικά του νέφους είτε της σύνδεσης του με τον εξωτερικό κόσμο. Η δικτυακή αυτή υπηρεσία φέρει την ονομασία Neutron. Το Neutron έκανε την πρώτη εμφάνιση του με την έκδοση Folsom του Openstack και το αρχικό της όνομα ήταν Quantum. Βασικό χαρακτηριστικό του Quantum ήταν ότι αποτελούσε ένα εντελώς ξεχωριστό εγχείρημα που μπορούσε να χρησιμοποιηθεί και μόνο του (SDN- Software Defined Network) . Σκοπός της δημιουργίας αυτής ήταν το γεγονός ότι ήθελαν να δημιουργήσουν ένα εντελώς ξεχωριστό λογισμικό για το γενικό οικοσύστημα του Openstack. Στην συνέχεια με την έκδοση του Havana Openstack μετονομάστηκε σε Neutron λόγω του trademark που είχε το όνομα Quantum.

Κατά την δημιουργία του Lab as a Service είχαμε την δυνατότητα επιλογής ανάμεσα σε δύο υπηρεσίες του διαχειρίζονταν το δικτυακό μέρος του Openstack. Εκτός από το Neutron, υπήρχε και η επιλογή του Nova Networking. Επιλέξαμε το Neutron, που παρότι η επιλογή μας δεν θα επηρέαζε την υλοποίηση μας , θεωρήσαμε πως είναι η καλύτερη σε οποιαδήποτε περίπτωση χρήσης του Openstack. Βασικό μειονέκτημα του Nova Networking είναι ότι έχει περιορισμούς στο 2ο επίπεδο (Data Link Layer). Επίσης το Nova Networking δεν μπορεί να εξυπηρετήσει παραπάνω από 4094 VLANs που αποτελεί το standard του 802.11q πρωτοκόλλου. Ένα άλλο μειονέκτημα του αποτελεί το γεγονός ότι δεν λειτουργεί πολύ καλά στον εξωτερικό κόσμο του οικοσυστήματος όσο λειτουργεί στο εσωτερικό του , μιας και δημιουργήθηκε για να λειτουργεί εντός Openstack μόνο.

Συνέπεια όλων των παραπάνω ήταν να δημιουργηθεί το Neutron που αποτελεί ένα εντελώς ξεχωριστό περιβάλλον.Το Neutron αποτελείται από κόμβους. Έτσι και εμείς υλοποιήσαμε έναν κόμβο εντός του περιβάλλοντος μας.

Ακολουθεί ένα παράδειγμα που θα μας δώσει να καταλάβουμε την δομή του Neutron και πως ενώνει τις διάφορες οντότητες εντός του περιβάλλοντος μας έτσι ώστε αυτές να μπορούν να επικοινωνούν μεταξύ τους.



5.9 Η Δομή της Υπηρεσίας Neutron

Στο παράδειγμα μας, παρατηρούμε ότι στο Network Node υφίστανται και τρέχουν τρεις διαφορετικοί Agents. Ο plugin-agent είναι συγκεκριμένος και βασίζεται στο Hardware ή ακόμα και στο Software του παρόχου. Υπάρχουν επίσης ένας layer-3 agent και ο DHCP-agent που βοηθούν στην παροχή IP και DHCP υπηρεσιών σε αυτήν την περιοχή. Συνεχίζοντας, στον Compute Node τρέχει ένας neutron-plugin-agent με σκοπό να υπάρχει επικοινωνία μεταξύ αυτού και του Network Node. Τέλος στον Controller Node, όπως βλέπουμε, υπάρχει ένας Neutron-Server. Σε αυτό το σημείο θα θέλαμε να επαναλάβουμε ότι το Neutron είναι μια υπηρεσία standalone, δηλαδή μπορεί να υπάρξει μόνη της εντός του περιβάλλοντος μας, που το βοηθά να επικοινωνεί με τον έξω κόσμο. Με αυτόν τον τρόπο και την μέθοδο χρήσης του Neutron μας δίνεται η δυνατότητα μεγαλύτερου ελέγχου του νέφους μας αλλά και η δυνατότητα εύκολης επεκτασιμότητας σε περίπτωση που το επιθυμούμε.

Για την εγκατάσταση και την ρύθμιση[47] του Neutron χρειάζεται να γίνουν κάποιες ενέργειες που προαπαιτούνται ώστε αυτό να λειτουργήσει, όπως ακριβώς συνέβη και με τις προηγούμενες υπηρεσίες. Έτσι θα πρέπει να δημιουργήσουμε μια βάση δεδομένων, τα στοιχεία ταυτοποίησης της υπηρεσίας και τα API Endpoints. Για το Neutron απαιτείται η εγκατάσταση ενός κόμβου ελέγχου, (Controller Node), ενός υπολογιστικού κόμβου(Compute Node) και ενός δικτυακού κόμβου (Networking Node). Όπως και στην ανάπτυξη του Nova έτσι και εδώ οι

τρεις κόμβοι φιλοξενούνται στην ίδια μηχανή, οπότε οι ρυθμίσεις έγιναν με βάση αυτήν.

Αρχικά συνδεόμαστε στον client της βάσης δεδομένων ως root με την εντολή:

```
$ mysql -u root –p
```

Αμέσως επόμενο βήμα είναι η δημιουργία της βάσης glance:

```
CREATE DATABASE neutron;
```

Μετά δίνουμε την κατάλληλη πρόσβαση στην βάση ως εξής:

```
GRANT ALL PRIVILEGES ON glance.* TO ' neutron '@'195.251.123.247' \
    IDENTIFIED BY ' neutron pass';
GRANT ALL PRIVILEGES ON glance.* TO ' neutron '@'%' \
    IDENTIFIED BY ' neutron pass';
```

Η επόμενη προαπαίτηση είναι η δημιουργία των στοιχείων ταυτοποίησης για το Glance. Για αυτόν τον σκοπό απαιτούνται κάποιες ενέργειες οι οποίες είναι:

1. Η δημιουργία του neutron χρήστη με το όρισμα:

```
$ keystone user-create --name neutron --pass neutron pass
```

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |                                  |
| enabled  |             True                 |
|    id    | 7fd67878dcd04d0393469ef825a7e005 |
|   name   |           neutron                |
| username |           neutron                |
+----------+----------------------------------+
```

2. Η προσθήκη του ρόλου admin στον χρήστη nova:

```
$ keystone user-role-add --user neutron --tenant service –role \ admin
```

3. Η δημιουργία της οντότητας της glance υπηρεσίας:

```
$ keystone service-create --name neutron --type network \
--description "OpenStack Networking"
```

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |        OpenStack Networking      |
|   enabled   |               True               |
|      id     | 6369ddaf99a447f3a0d41dac5e342161 |
|     name    |              neutron             |
|     type    |              network             |
+-------------+----------------------------------+
```

4. Η δημιουργία των Glance API Endpoints

```
$ keystone endpoint-create \
--service-id $(keystone service-list | awk '/ network / \
  {print$2}')
  --publicurl http://195.251.123.247:9696 \
  --internalurl http:// 195.251.123.247:9696 \
  --adminurl http:// 195.251.123.247:9696 \
  --region regionOne
```

```
+-------------+----------------------------------=-------------+
|   Property  |                   Value                        |
+-------------+------------------------------------------------+
|   adminurl  |        http:// 195.251.123.247:9696            |
|      id     |        fa18b41938a94bf6b35e2c152063ee21        |
| internalurl |        http:// 195.251.123.247:9696            |
|  publicurl  |        http:// 195.251.123.247:9696            |
|    region   |                  regionOne                     |
|  service_id |     6369ddaf99a447f3a0d41dac5e34216            |
+-------------+------------------------------------------------+
```

Στην συνέχεια για την εγκατάσταση και την παραμετροποίηση του Neutron, πρώτα έγινε η λήψη των πακέτων που ήταν απαραίτητα:

```
# apt-get install neutron-server neutron-plugin-ml2 \
  python-neutronclient
```

Μετά την λήψη των παραπάνω πακέτων, σειρά είχε η ρύθμιση του συστατικού που λέγεται Networking Server. Οι ρυθμίσεις που υπάγονται σε αυτό

είναι η βάση δεδομένων, ο μηχανισμός αυθεντικοποίησης, ο message broker, οι ειδοποιήσεις για αλλαγές στην τοπολογία του δικτύου και τα plug-ins.

Πρώτα ρυθμίστηκε το αρχείο /etc/neutron/neutron.conf και έγιναν οι παρακάτω αλλαγές:

a. Στο τμήμα [database] , ρυθμίστηκε η πρόσβαση στην βάση δεδομένων:

```
1 [database]
2 . . .
3 connection=mysql://neutron:neutronpass@195.251.123.247/neutron
```

b. Στο [DEFAULT] , ρυθμίστηκε η πρόσβαση του RabbitMQ Message Broker:

```
1 [DEFAULT]
2 . . .
3 rpc_backend = rabbit
4 rabbit_host = 195.251.123.247
5 rabbit_password = rabbitpass
```

c. Στο [DEFAULT]και [keystone_authtoken]τμήμα, ρυθμίστηκε η πρόσβαση στην υπηρεσία ταυτοποίησης Keystone:

```
1 [DEFAULT]
2 . . .
3 auth_strategy = keystone
4 [keystone_authtoken]
5 . . .
6 auth_uri = http://195.251.123.247:5000/v2.0
7 identity_uri = http://195.251.123.247:35357
8 admin_tenant_name = service
```

```
9       admin_user = neutron

10      admin_password = neutronpass
```

d. Στο [DEFAULT] τμήμα, ενεργοποιήθηκε το Modular Layer 2 (ML2) plug-in, η υπηρεσία δρομολόγησης και το overlapping IP διευθύνσεων:

```
1       [DEFAULT]
2       . . .
3       core_plugin = ml2
4       service_plugins = router
5       allow_overlapping_ips = True
```

e. Στο [DEFAULT] τμήμα, ρυθμίστηκε το Networking έτσι ώστε να ειδοποιεί το Compute για τυχόν αλλαγές στην τοπολογία:

```
1       [DEFAULT]
2       . . .
3       notify_nova_on_port_status_changes = True
4       notify_nova_on_port_data_changes = True
5       nova_url = http://195.251.123.247:8774/v2
6       nova_admin_auth_url = http:// 195.251.123.247:35357/v2.0
7       nova_region_name = regionOne
8       nova_admin_username = nova
9       nova_admin_tenant_id = f7275ec2ceb4d71bad86fc414449bf
10      nova_admin_password = novapass
```

Επόμενο βήμα ήταν η ρύθμιση του ML2 plug-in. Το ML2 χρησιμοποιεί τον Open vSwitch (OVS) μηχανισμό (agent) για την κατασκευή του εικονικού δικτυακού Framework των instances. Όμως, ο controller node δεν χρειάζεται το OVS καθώς δεν διαχειρίζεται την δικτυακή κίνηση των instances.

Ρυθμίσαμε το αρχείο /etc/neutron/plugins/ml2/ml2_conf.ini ως εξής:

a. Στο [ml2] τμήμα του αρχείου, ενεργοποιήσαμε τους δικτυακούς τύπους οδηγών flat και Generic Routing Encapsulation (GRE), τα GRE tenant networks και τον OVS οδηγό:

```
1       [ml2]
2       . . .
3       type_drivers = flat,gre
4       tenant_network_types = gre
5       mechanism_drivers = openvswitch
```

b. Στο τμήμα [ml2_type_gre] του αρχείου, ρυθμίσαμε το εύρος του tunnel identifier:

```
1 [ml2_type_gre]
2 . . .
3 tunnel_id_ranges = 1:1000
```

c. Στο [securitygroup] τμήμα του αρχείου, ενεργοποιήσαμε τα security groups, το ipset και ρυθμίσαμε τα OVS iptables και τον οδηγό του τοίχους προστασίας:

```
1 [securitygroup]
2 . . .
3 enable_security_group = True
4 enable_ipset = True
5 firewall_driver=neutron.agent.linux.iptables_firewall.
  OVSHybridIptablesFirewallDriver
```

Στην συνέχεια, σειρά είχε η ρύθμιση του Compute με σκοπό να μπορεί να χρησιμοποιεί το Networking. Προεπιλογή των πακέτων μετά την λήψη είναι η χρήση του legacy networking από το Compute. Για αυτόν τον λόγο έπρεπε να γίνει η αλλαγή έτσι ώστε να γίνεται η διαχείριση των δικτύων μέσω του Neutron.

Ρυθμίσαμε το αρχείο /etc/nova/nova.conf και κάναμε τις παρακάτω προσθήκες και αλλαγές:

a. Στο [DEFAULT] τμήμα, ρυθμίσαμε τους οδηγούς και τα APIs:

```
1 [DEFAULT]
2 . . .
3 network_api_class = nova.network.neutronv2.api.API
4 security_group_api = neutron
5 linuxnet_interface_driver=nova.network.linux_net.
  LinuxOVSInterfaceDriver
6 firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

b. Στο τμήμα [neutron], ρυθμίσαμε τις παραμέτρους πρόσβασης ως εξής:

```
1 [neutron]
2 . . .
3 url = http://195.251.123.247:9696
4 auth_strategy = keystone
```

```
5       admin_auth_url = http://195.251.123.247:35357/v2.0
6       admin_tenant_name = service
7       admin_username = neutron
8       admin_password = neutronpass
```

Στο σημείο αυτό για να γίνουν οι αλλαγές και να ισχύουν κατά την λειτουργία του συστήματος επανεκκινήσαμε τις υπηρεσίες Nova και Neutron:

```
# service nova-api restart
# service nova-scheduler restart
# service nova-conductor restart
# service neutron-server restart
```

Για την επαλήθευση όλης της παραπάνω διαδικασίας ελέγξαμε την λίστα με τα extensions του neutron-server εν ώρα λειτουργίας:

```
$ neutron ext-list
```

```
+-----------------------+-----------------------------------------------+
| alias                 | name                                          |
+-----------------------+-----------------------------------------------+
| security-group        | security-group                                |
| l3_agent_scheduler    | L3 Agent Scheduler                            |
| ext-gw-mode           | Neutron L3 Configurable external gateway mode |
| binding               | Port Binding                                  |
| provider              | Provider Network                              |
| agent                 | agent                                         |
| quotas                | Quota management support                      |
| dhcp_agent_scheduler  | DHCP Agent Scheduler                          |
| l3-ha                 | HA Router extension                           |
| multi-provider        | Multi Provider Network                        |
| external-net          | Neutron external network                      |
| router                | Neutron L3 Router                             |
| allowed-address-pairs | Allowed Address Pairs                         |
| extraroute            | Neutron Extra Route                           |
| extra_dhcp_opt        | Neutron Extra DHCP opts                       |
| dvr                   | Distributed Virtual Router                    |
+-----------------------+-----------------------------------------------+
```

Μετά τον controller node, σειρά είχε η ρύθμιση του δικτυακού κόμβου που κυρίως διαχειρίζεται την εσωτερική και εξωτερική δρομολόγηση και DHCP υπηρεσίες για εικονικά δίκτυα.

Προαπαιτούμενες ενέργειες πριν την εγκατάσταση του Openstack Networking είναι κάποιες αλλαγές που έγιναν σε παραμέτρους του δικτυακού kernel στο αρχείο /etc/sysctl.conf. Οι αλλαγές αυτές είναι οι εξής:

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=0
```

```
net.ipv4.conf.default.rp_filter=0
```

Η εφαρμογή των αλλαγών γίνεται με την εντολή:

```
# sysctl -p
```

Για την εγκατάσταση των συστατικών του Networking λήφθηκαν τα παρακάτω πακέτα με την εντολή:

```
# apt-get install neutron-plugin-ml2 \
  neutron-plugin-openvswitch agent neutron-l3-agent \
  neutron-dhcp-agent
```

Στο σημείο αυτό επειδή και η υλοποίηση μας βασίζεται σε έναν κόμβο οι ρυθμίσεις που έπρεπε να γίνουν στα αρχεία /etc/neutron/neutron.conf και /etc/neutron/plugins/ml2/ml2_conf.ini του Network Node, έχουν ήδη γίνει παραπάνω στις ρυθμίσεις που αφορούσαν τον Controller Node.

Για την ρύθμιση του Layer-3 (L3) agent που παρέχει υπηρεσίες δρομολόγησης εικονικών δικτύων έγιναν οι εξής αλλαγές:

Στο αρχείο /etc/neutron/l3_agent.ini και πιο συγκεκριμένα στο τμήμα `[DEFAULT]`, ρυθμίσαμε τον οδηγό, ενεργοποιήσαμε τα namespaces των δικτύων, ρυθμίσαμε την δικτυακή γέφυρα του εξωτερικού δικτύου και ενεργοποιήσαμε την διαγραφή των namespaces στον δρομολογητή:

```
1  [DEFAULT]
2  . . .
3  interface_driver=neutron.agent.linux.interface.
   OVSInterfaceDriver
4  use_namespaces = True
5  external_network_bridge = br-ex
6  router_delete_namespaces = True
```

Αφού έγινε η ρύθμιση του L3 agent, σειρά είχε ο DHCP agent, ο οποίος παρέχει υπηρεσίες DHCP για τα εικονικά δίκτυα.

Στο αρχείο /etc/neutron/dhcp_agent.ini έγιναν οι παρακάτω αλλαγές:

a. Στο τμήμα του αρχείου, ρυθμίστηκαν οι οδηγοί, ενεργοποιήθηκαν τα namespaces των δικτύων και η διαγραφή των namespaces για το DHCP:

```
1  [DEFAULT]
2  . . .
3  interface_driver=neutron.agent.linux.interface.
   OVSInterfaceDriver
4  dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
5  use_namespaces = True
6  dhcp_delete_namespaces = True
```

Επόμενο βήμα ήταν η ρύθμιση του agent για τα μεταδεδομένα, ο οποίος παρέχει πληροφορίες ρυθμίσεων όπως π.χ. στοιχεία ταυτοποίησης στα instances. Στο αρχείο /etc/neutron/metadata_agent.ini έγιναν οι ακόλουθες προσθήκες:

a. Στο [DEFAULT] τμήμα, ρυθμίστηκαν οι παράμετροι πρόσβασης:

```
1  [DEFAULT]
2  . . .
3  auth_url = http://195.251.123.247:5000/v2.0
4  auth_region = regionOne
5  admin_tenant_name = service
6  admin_user = neutron
7  admin_password = neutronpass
```

b. Στο [DEFAULT] τμήμα, ρυθμίστηκε ο Host των μεταδεδομένων:

```
1  [DEFAULT]
2  . . .
3  nova_metadata_ip = 195.251.123.247
```

c. Στο [DEFAULT] τμήμα, ρυθμίστηκε το μεταδεδομένο proxy shared secret:

```
1  [DEFAULT]
2  . . .
3  metadata_proxy_shared_secret = metadatasecret
```

Στο αρχείο /etc/nova/nova.conf προστέθηκαν οι παρακάτω αλλαγές:

Στο [neutron] τμήμα του αρχείου, ενεργοποιήθηκε ο proxy των μεταδεδομένων και ρυθμίστηκε το secret πεδίο του:

```
1  [neutron]
```

```
2       . . .
3       service_metadata_proxy = True
4       metadata_proxy_shared_secret = metadatasecret
```

Τελευταία υπηρεσία προς ρύθμιση είναι η Open vSwitch (OVS). Αυτή η υπηρεσία παρέχει το framework για την εικονική δικτύωση των instances. Η integration γέφυρα br-int είναι αρμόδια για την κίνηση των instances του εσωτερικού δικτύου εντός του OVS. Η εξωτερική γέφυρα br-ex είναι αρμόδια για την κίνηση των instances του εξωτερικού δικτύου εντός του OVS. Η εξωτερική γέφυρα απαιτεί ένα port στο φυσικό εξωτερικό interface του δικτύου για να παρέχει στα instances εξωτερική δικτύωση. Στην ουσία, αυτό το port συνδέει τα εικονικά και φυσικά εξωτερικά δίκτυα στο περιβάλλον μας.

Αρχικά , επανεκκινήσαμε την υπηρεσία OVS:

```
# service openvswitch-switch restart
```

Αμέσως μετά προσθέσαμε την εξωτερική γέφυρα:

```
# ovs-vsctl add-br br-ex
```

Προσθέσαμε, ένα port στην εξωτερική γέφυρα το οποίο συνδέεται με το interface του εξωτερικού, φυσικού δικτύου:

```
# ovs-vsctl add-port br-ex eth0
```

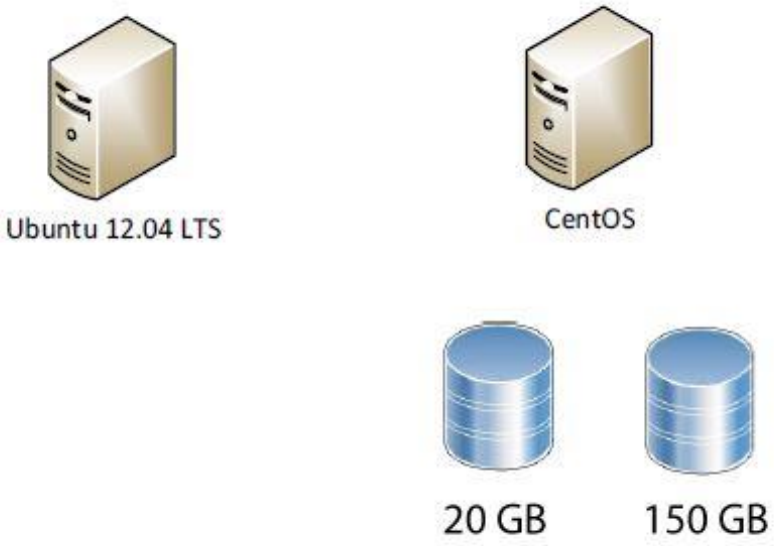
Για την ολοκλήρωση και την επαλήθευση της διαδικασίας, έγινε επανεκκίνηση των υπηρεσιών δικτύωσης και δόθηκε εντολή για την προβολή της λίστας με τους agents του Neutron.

```
# service neutron-plugin-openvswitch-agent restart
# service neutron-l3-agent restart
# service neutron-dhcp-agent restart
# service neutron-metadata-agent restart
$ neutron agent-list
```

5.4.5 Cinder

Το Cinder είναι και αυτή βασική υπηρεσία του Openstack. Σκοπός του είναι να διαχειρίζεται τα αποθηκευτικά μέσα που διαθέτει ο υπολογιστής στον οποίο έχουμε στήσει το περιβάλλον μας.

Χρησιμοποιώντας την υπηρεσία του Cinder, έχουμε την δυνατότητα να δημιουργήσουμε block volumes, δηλαδή αποθηκευτικούς χώρους που κάθε μία αποτελεί μια ξεχωριστή οντότητα. Στην συνέχεια, αυτούς τους αποθηκευτικούς χώρους έχουμε την δυνατότητα να τους ορίσουμε ως αποθηκευτικές μονάδες σε instances που έχουμε ήδη δημιουργήσει και είναι σε λειτουργία. Αυτά τα block volumes έχουν το θετικό χαρακτηριστικό ότι δεν καταστρέφονται στην περίπτωση που καταργήσουμε ένα instance που δεν επιθυμούμε να λειτουργεί ή όταν το διαγράψουμε. Θετική συνέπεια του παραπάνω χαρακτηριστικού είναι η δυνατότητα να μπορούμε να μεταφέρουμε δεδομένα που τα χρειαζόμαστε στην περίπτωση που θέλουμε να αλλάξουμε περιβάλλον εργασίας εντός του νέφους μας.



5.10 Cinder και φορητότητα των αποθηκευτικών χώρων

Ας υποθέσουμε ότι ο διαχειριστής του συστήματος θέλει να τρέξει ένα CentOS περιβάλλον με σκοπό να δημιουργήσει μια εφαρμογή. Για τις ανάγκες του χρειάζεται έναν μόνιμο αποθηκευτικό χώρο, έτσι χρησιμοποιώντας το Cinder δημιουργεί έναν χώρο 20GB τον οποίο στην συνέχεια επισυνάπτει στο CentOS instance που δημιούργησε με την βοήθεια του Nova. Μετέπειτα, αποφασίζει πως

τα 20GB που είχε ορίσει αρχικά δεν είναι επαρκή. Έτσι αποφασίζει και δημιουργεί ακόμα 150GB αποθηκευτικού χώρου το οποίο και αυτό το επισυνάπτει στο CentOS Instance.Σε μεταγενέστερη φάση και μετά από χρήση της εφαρμογής του, συμπεραίνει ότι αυτή θα είναι πιο αποδοτική και πιο εύκολα διαχειρίσιμη αν είναι εγκατεστημένη σε περιβάλλον Ubuntu. Το μόνο που έχει να κάνει λοιπόν είναι να αποκολλήσει τις 2 αποθηκευτικές μονάδες από το CentOS instance , να το τερματίσει και στην συνέχεια να επικολλήσει τα block volumes στο Ubuntu instance αφού πρώτα φυσικά το δημιουργήσει και το βάλει σε λειτουργία. Έτσι θα μεταφέρει την εφαρμογή του στην νέα του πλατφόρμα χωρίς να αλλάξουν τα δεδομένα που υπήρχαν όταν αυτή βρισκόταν στην αρχική πλατφόρμα.

Είναι εύκολο να κατανοήσουμε πως λειτουργούν οι αποθηκευτικές μονάδες του Cinder που εμείς δημιουργούμε. Είναι αποθηκευμένες εντός του αποθηκευτικού περιβάλλοντος του νέφους μας. Είναι διαθέσιμα προς χρήση από τα instances μας, χρησιμοποιώντας την πολύ γνωστή iSCSI τεχνολογία. Τα εργαλεία που χρησιμοποιούμε για να διαχειριστούμε τους αποθηκευτικούς χώρους που δημιουργούμε είναι η επισύναψη και η αποκόλληση τους από τα Nova instances που τρέχουμε. Επίσης μας δίνονται οι επιλογές του clone και snapshot δημιουργώντας αντίγραφο τους για μεταγενέστερη χρήση ή αλλαγή. Ακόμα ένα βασικό χαρακτηριστικό τους είναι ότι μας δίνεται η δυνατότητα του boot ενός λειτουργικού συστήματος από volume ή snapshot.

Η αρχική διαδικασία της εγκατάστασης[48] του Cinder είναι παρόμοια των υπολοίπων βασικών υπηρεσιών του Openstack. Αυτό περιλαμβάνει την δημιουργία μιας βάσης δεδομένων, μιας υπηρεσίας στοιχείων ταυτοποίησης και των API Endpoints.

```
$ mysql -u root –p
```

Αμέσως επόμενο βήμα είναι η δημιουργία της βάσης Cinder:

```
CREATE DATABASE cinder;
```

Μετά δίνουμε την κατάλληλη πρόσβαση στην βάση ως εξής:

```
GRANT ALL PRIVILEGES ON cinder.* TO ' cinder '@'195.251.123.247' \
    IDENTIFIED BY ' cinder pass';
GRANT ALL PRIVILEGES ON cinder.* TO ' cinder '@'%' \
    IDENTIFIED BY ' cinder pass';
```

Η επόμενη προαπαίτηση είναι η δημιουργία των στοιχείων ταυτοποίησης για το Cinder. Για αυτόν τον σκοπό απαιτούνται κάποιες ενέργειες οι οποίες είναι:

1. Η δημιουργία του neutron χρήστη με το όρισμα:

$ keystone user-create --name cinder --pass cinder*pass*

```
+----------+----------------------------------+
| Property |              Value               |
+----------+----------------------------------+
|  email   |                                  |
| enabled  |             True                 |
|    id    | 881ab2de4f7941e79504a759a83308be |
|   name   |           cinder                 |
| username |           cinder                 |
+----------+----------------------------------+
```

2. Η προσθήκη του ρόλου admin στον χρήστη nova:

$ keystone user-role-add --user cinder --tenant service –role \ admin

3. Η δημιουργία της οντότητας της glance υπηρεσίας:

$ keystone service-create --name cinder --type volume \
--description "OpenStack Block Storage"

```
+-------------+-----------------------------------+
|   Property  |               Value               |
+-------------+-----------------------------------+
| description |        OpenStack Block Storage    |
|   enabled   |                True               |
|      id     |  6369ddaf99a447f3a0d41dac5e342161 |
|     name    |               cinder              |
|     type    |               volume              |
+-------------+-----------------------------------+
```

$ keystone service-create --name cinderv2 --type volumev2 \
--description "OpenStack Block Storage"

```
+-------------+----------------------------------+
|   Property  |              Value               |
+-------------+----------------------------------+
| description |     OpenStack Block Storage      |
|   enabled   |               True               |
|      id     | 16e038e449c94b40868277f1d801edb5 |
|     name    |             cinderv2             |
|     type    |             volumev2             |
+-------------+----------------------------------+
```

Δημιουργήθηκαν δύο διαφορετικές οντότητες για το Cinder διότι η κάθε μία υποστηρίζει διαφορετική έκδοση του API. Υπάρχουν οι εκδόσεις 1 και 2.

4. Η δημιουργία των Glance API Endpoints

```
$ keystone endpoint-create \
--service-id $(keystone service-list | awk '/ volume / \      {print$2}')
 --publicurl http://195.251.123.247:8776/v1/%\(tenant_id\)s \
 --internalurl http://195.251.123.247:8776/v1/%\(tenant_id\)s \
 --adminurl http://195.251.123.247:8776/v1/%\(tenant_id\)s \
 --region regionOne
```

```
+-------------+----------------------------------=-------------+
|   Property  |                   Value                        |
+-------------+------------------------------------------------+
|   adminurl  |http:// 195.251.123.247:8776/v1/%(tenant_id)s |
|      id     |        d1b7291a2d794e26963b322c7f2a55a4       |
| internalurl |http:// 195.251.123.247:8776/v1/%(tenant_id)s |
|  publicurl  |http:// 195.251.123.247:8776/v1/%(tenant_id)s |
|    region   |                  regionOne                    |
|  service_id |        1e494c3e22a24baaafcaf777d4d467e       |
+-------------+------------------------------------------------+
```

```
$ keystone endpoint-create \
--service-id $(keystone service-list | awk '/ volumev2 / \      {print$2}')
 --publicurl http://195.251.123.247:8776/v2/%\(tenant_id\)s \
 --internalurl http://195.251.123.247:8776/v2/%\(tenant_id\)s \
 --adminurl http://195.251.123.247:8776/v2/%\(tenant_id\)s \
         --region regionOne
```

```
+-------------+----------------------------------=-------------+
|   Property  |                   Value                         |
+-------------+-------------------------------------------------+
|   adminurl  |http:// 195.251.123.247:8776/v2/%(tenant_id)s |
|      id     |       097b4a6fc8ba44b4b10d4822d2d9e076        |
| internalurl |http:// 195.251.123.247:8776/v2/%(tenant_id)s |
|  publicurl  |http:// 195.251.123.247:8776/v2/%(tenant_id)s |
|    region   |                  regionOne                    |
|  service_id |       16e038e449c94b40868277f1d801edb         |
+-------------+-------------------------------------------------+
```

Στην συνέχεια για την εγκατάσταση και την παραμετροποίηση του Cinder, πρώτα έγινε η λήψη των πακέτων που ήταν απαραίτητα:

```
# apt-get install cinder-api cinder-scheduler python-cinderclient
```

Πρώτα ρυθμίστηκε το αρχείο /etc/cinder/cinder.conf και έγιναν οι παρακάτω αλλαγές:

a. Στο τμήμα [database] , ρυθμίστηκε η πρόσβαση στην βάση δεδομένων:

```
1 [database]
2 . . .
3 connection=mysql://cinder:cinderpass@195.251.123.247/cinder
```

b. Στο [DEFAULT] , ρυθμίστηκε η πρόσβαση του RabbitMQ Message Broker:

```
1 [DEFAULT]
2 . . .
3 rpc_backend = rabbit
4 rabbit_host = 195.251.123.247
5 rabbit_password = rabbitpass
```

c. Στο [DEFAULT]και [keystone_authtoken]τμήμα, ρυθμίστηκε η πρόσβαση στην υπηρεσία ταυτοποίησης Keystone:

```
1 [DEFAULT]
2 . . .
```

```
3   auth_strategy = keystone

4   [keystone_authtoken]

5   . . .

6   auth_uri = http://195.251.123.247:5000/v2.0

7   identity_uri = http://195.251.123.247:35357

8   admin_tenant_name = service

9   admin_user = cinder

10  admin_password = cinderpass
```

d. Στο [DEFAULT] τμήμα, ρυθμίστηκε η επιλογή my_ip ώστε να είναι η IP διεύθυνση του management interface του κόμβου ελέγχου.

```
1   [DEFAULT]
2   . . .
3   my_ip = 195.251.123.247
```

Για την ολοκλήρωση της διαδικασίας εγκατάστασης του cinder στον κόμβο ελέγχου επανεκκινήσαμε τις υπηρεσίες Cinder:

```
# service cinder-scheduler restart
# service cinder-api restart
```

Αμέσως επόμενο βήμα ήταν η παραμετροποίηση του κόμβου αποθήκευσης. Για τον σκοπό αυτό δημιουργήσαμε ένα διαφορετικό partition του φυσικού αποθηκευτικού μέσου για να εξυπηρετήσει τον σκοπό αυτό.

Για την διαδικασία του partitioning χρησιμοποιήσαμε το LVM (Logical Volume Manager). Η λήψη των απαραίτητων πακέτων έγινε με την εντολή:

```
# apt-get install lvm2
```

Στην συνέχεια δημιουργήσαμε τον φυσικό αποθηκευτικό χώρο /dev/sdb1 με το LVM :

```
# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
```

Μετά την δημιουργία, με την βοήθεια του LVM δημιουργήσαμε ένα group από αποθηκευτικούς χώρους με όνομα `cinder-volumes`:

```
# vgcreate cinder-volumes /dev/sdb1
  Volume group "cinder-volumes" successfully created
```

Η υπηρεσία Cinder δημιουργεί λογικούς αποθηκευτικούς χώρους σε αυτό το group. Μόνο τα instances έχουν πρόσβαση σε αυτούς τους αποθηκευτικούς χώρους. Ωστόσο, το λειτουργικό σύστημα που βρίσκεται στο υπόβαθρο διαχειρίζεται τις συσκευές που σχετίζονται με αυτούς τους αποθηκευτικούς χώρους. Σαν προεπιλογή, το εργαλείο σάρωσης των LVM αποθηκευτικών χώρων, σαρώνει τον /dev κατάλογο για συσκευές αποθήκευσης που περιέχουν αποθηκευτικούς χώρους δεδομένων. Εάν, τα tenants χρησιμοποιούν το LVM στους αποθηκευτικούς χώρους τους, το εργαλείο σάρωσης τους ανιχνεύει και προσπαθεί να τους περάσει στην cache μνήμη. Αυτό το γεγονός μπορεί να προκαλέσει πολλά προβλήματα τόσο με το υποκείμενο λειτουργικό σύστημα όσο και με τα tenants. Έτσι πρέπει να ρυθμίσουμε το LVM να σαρώνει μόνο τις συσκευές που περιέχουν cinder-volume αποθηκευτικά groups. Αυτή η ρύθμιση γίνεται κάνοντας τις απαραίτητες αλλαγές στο αρχείο /etc/lvm/lvm.conf ως εξής:

Στο τμήμα devices, προσθέσαμε ένα φίλτρο για να γίνονται δεκτές συσκευές `/dev/sdb` και όλες οι άλλες συσκευές να απορρίπτονται:

```
1  devices {
2  . . .
3  filter = [ "a/sdb/", "r/.*/"]
```

Στην συνέχεια για την εγκατάσταση και την παραμετροποίηση του Cinder, εγκαταστήσαμε τα πακέτα:

```
# apt-get install cinder-volume python-mysqldb
```

Για την παραμετροποίηση, χρειάστηκε μια περαιτέρω αλλαγή στο αρχείο /etc/cinder/cinder.conf καθώς η ανάπτυξη μας έγινε σε έναν κόμβο:

Στο τμήμα [DEFAULT] του αρχείου, ρυθμίσαμε την τοποθεσία του Glance:

```
1       [DEFAULT]
2       . . .
3       glance_host = 195.251.123.247
```

Για να εφαρμοστούν οι παραπάνω αλλαγές και να τεθεί σε λειτουργία το σύστημα επανεκκινήσαμε μερικές υπηρεσίες:

```
# service tgt restart
# service cinder-volume restart
```

Τέλος για να επαληθεύσουμε την λειτουργία της παραπάνω υπηρεσίας αποκτήσαμε αρχικά πρόσβαση στο admin-only CLI με την εντολή:

```
$ source admin-openrc.sh
```

Στην συνέχεια δώσαμε όρισμα για να δούμε την λίστα των services του cinder:

```
$ cinder service-list
+------------------+-----------+------+---------+-------+---------------+
|      Binary      |    Host   | Zone |  Status | State |Disabled Reason|
+------------------+-----------+------+---------+-------+---------- ----+
| cinder-scheduler | controller| nova | enabled |   up  |    None       |
|  cinder-volume   |   block1  | nova | enabled |   up  |    None       |
+------------------+-----------+------+---------+-------+---------------+
```

Επίσης σαν demo χρήστης με την εντολή:

```
$ source demo-openrc.sh
```

Δημιουργήσαμε έναν αποθηκευτικό χώρο 1 GB:

```
$ cinder create --display-name demo-volume1 1
```

```
+---------------------+--------------------------------------+
|       Property      |                Value                 |
+---------------------+--------------------------------------+
|     attachments     |                  []                  |
|  availability_zone  |                 nova                 |
|       bootable      |                false                 |
|      created_at     |      2014-12-5T11:48:20.870239       |
| display_description |                 None                 |
|     display_name    |             demo-volume1             |
|      encrypted      |                False                 |
|          id         | 158bea89-07db-4ac2-8115-66c0d6a4bb48 |
|       metadata      |                  {}                  |
|         size        |                  1                   |
|     snapshot_id     |                 None                 |
|     source_volid    |                 None                 |
|        status       |              creating                |
|     volume_type     |                 None                 |
+---------------------+--------------------------------------+
```

Μετά την δημιουργία του παραπάνω αποθηκευτικού χώρου επαληθεύσαμε την δημιουργία του με την παρακάτω εντολή:

```
$ cinder list
```

```
+--------------------------------------+-----------+--------------+-----+
|                  ID                  |   Status  | Display Name | Size|
+--------------------------------------+-----------+--------------+-----+
| 158bea89-07db-4ac2-8115-66c0d6a4bb48 | available | demo-volume1 |  1  |
+--------------------------------------+-----------+--------------+-----+
```

5.4.6 Horizon (Dashboard)

Τελευταίο κομμάτι του συστήματος μας και το οποίο αποτελεί και αυτό μέρος του πυρήνα του είναι η υπηρεσία Horizon. Μία από τις πολλές προκλήσεις που έχουν να αντιμετωπίσουν οι κατασκευαστές που δημιουργούν κάθε είδους περιβάλλοντα , είτε αυτά είναι νέφους είτε όχι , είναι να μπορέσουν να παρέχουν μια καλή εμπειρία χρήσης του περιβάλλοντος στους πελάτες τους. Σκοπός τους είναι οι πελάτες να μπορούν να χρησιμοποιούν το περιβάλλον και τις υπηρεσίες του με ευχέρεια και αποτελεσματικότητα το οποίο, στο μέλλον, μπορεί να οδηγήσει στην διάδοση και την διεύρυνση του προϊόντος.

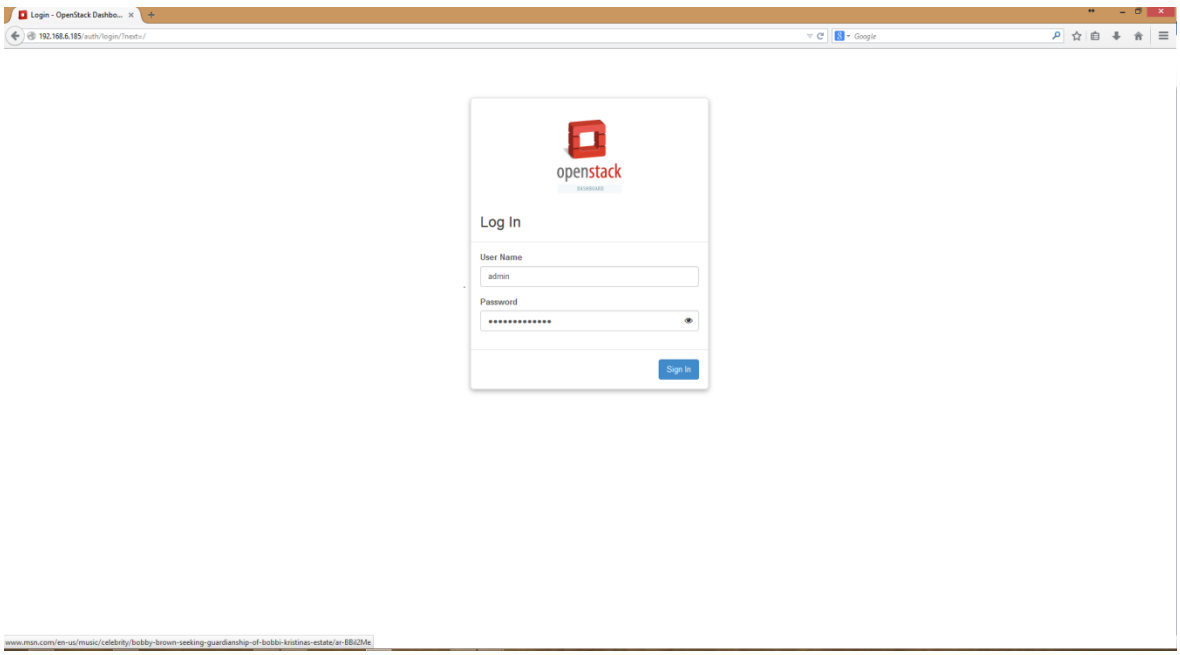
Στο Openstack η παροχή της διεπαφής στον χρήστη γίνεται με την υπηρεσία Horizon ή όπως συνηθίζεται να λέγεται Openstack Dashboard. Ο σκοπός της ύπαρξης μιας διεπαφής χρήστη (User Interface – UI) είναι να κατορθώνει να παρέχει στον χρήστη σημαντικές πληροφορίες για το συμβαίνει εντός του περιβάλλοντος και να του δίνει την δυνατότητα με εύκολο τρόπο να το διαχειρίζεται και να το παραμετροποιεί κατά το δοκούν. Επίσης είναι εύκολα κατανοητό ότι η διεπαφή του διαχειριστή θα πρέπει να περιέχει περισσότερες επιλογές και κουμπιά για να μπορεί να διαχειρίζεται και να συντηρεί το περιβάλλον.

Επομένως , ο απόλυτος σκοπός του Openstack είναι να μπορεί να παρέχει έναν οπτικό πίνακα ελέγχου για οποιονδήποτε χρήστη επιθυμεί να το χρησιμοποιήσει. Το Horizon αποτελεί μια πολύ απλή διεπαφή , όπου ο χρήστης μπορεί να την αυτοδιαχειριστεί με ευκολία. Πλεονέκτημα του Horizon επίσης αποτελεί το γεγονός ότι είναι προσβάσιμο και διαχειρίσιμο από οποιοδήποτε μέσο. Έτσι μπορούμε να έχουμε πρόσβαση μέσω κινητής συσκευής, ταμπλέτας, φορητού υπολογιστή και γενικά οποιοδήποτε τερματικού που έχει πρόσβαση στο διαδίκτυο.

Το παραπάνω χαρακτηριστικό, δηλαδή να μπορούμε να έχουμε πρόσβαση από πολλαπλές και διαφορετικές πλατφόρμες αποτελούσε αρχή σχεδίασης του Horizon. Επίσης, πάντα είναι προσβάσιμο ως ένα απλό html περιβάλλον όπου περιέχει ένα τεράστιο πλήθος δεδομένων και υπηρεσιών στο εσωτερικό του. Τέλος, έχει το χαρακτηριστικό ότι φιλοξενεί τόσο την πλατφόρμα της

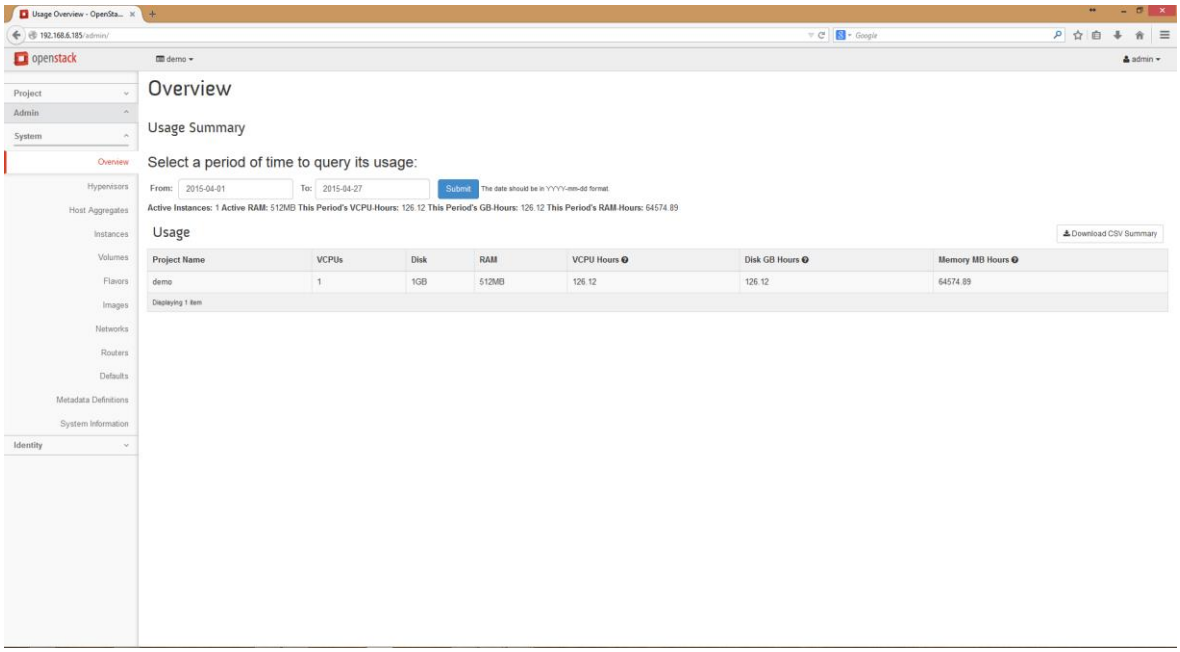
αυθεντικοποίησης όσο και της εξουσιοδότησης , έτσι δένει αρμονικά με το περιβάλλον του Keystone που ήδη αναφέραμε.

Στην συνέχεια της ενότητας αυτής ακολουθεί μια περιήγηση στο front-end μέρος του Horizon , μέσω εικόνων για να μας δώσει μια καλύτερη ιδέα και να έχουμε μια σφαιρική άποψη για την υπηρεσία αυτήν. Να προστεθεί ότι αποτελεί παρουσίαση της διεπαφής του διαχειριστή του συστήματος και πρόσβαση σε αυτήν έχουμε μέσω οποιουδήποτε φυλλομετρητή πληκτρολογώντας την IP του server μας.



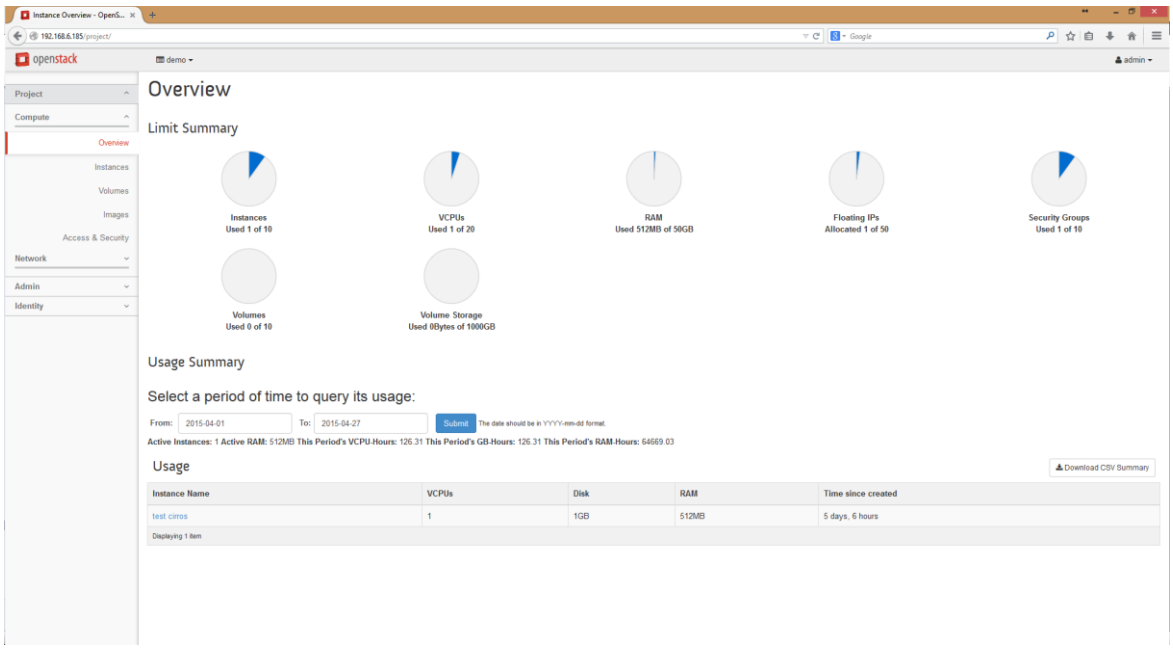
5.11 Η είσοδος στο Dashboard του περιβάλλοντος μας

Στην περίπτωση μας θέλουμε να δούμε τις λειτουργίες που είναι διαθέσιμες στον διαχειριστή του συστήματος, για αυτόν τον λόγο λοιπόν συνδεόμαστε στο περιβάλλον νέφους μας με τα στοιχεία ταυτοποίησης του διαχειριστή.



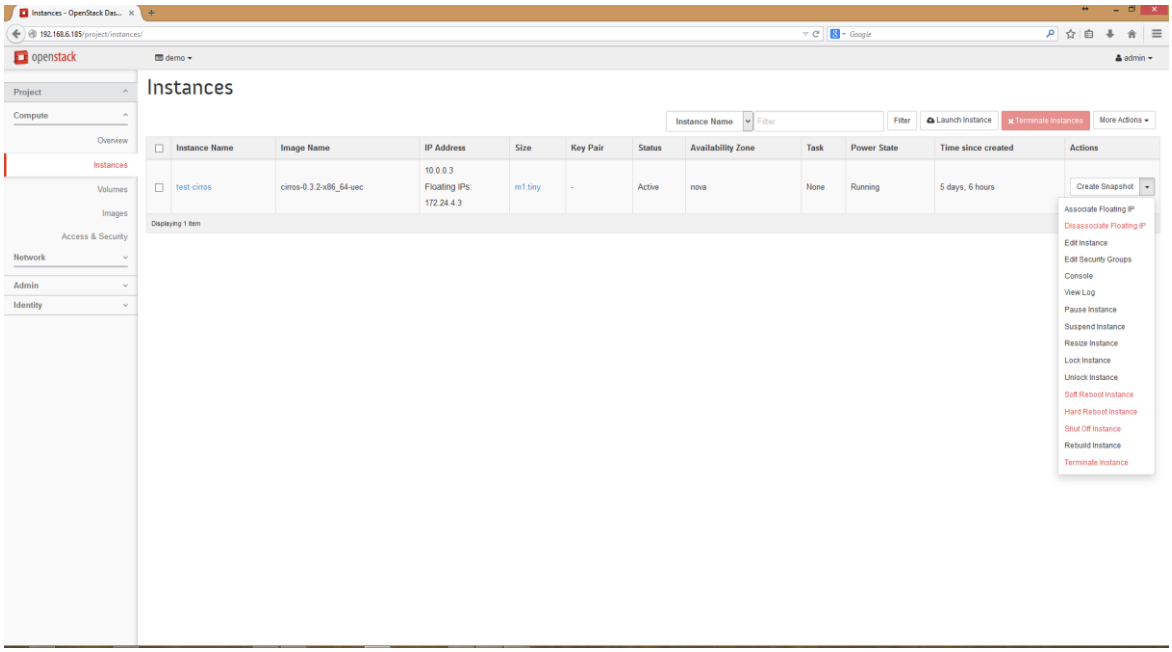
5.12 Η Κεντρική Προεπισκόπηση του Horizon

Όπως μπορούμε να δούμε στην Εικόνα 5.12 το Openstack Dashboard αποτελεί μια πολύ απλή και με μεγάλη ευκρίνεια διεπαφή χρήστη. Το μεγάλο πλεονέκτημα της κεντρικής προεπισκόπησης είναι ότι περιέχει τα μενού όλων των υπηρεσιών του Openstack για γρήγορη πλοήγηση σε αυτά. Βέβαια, το μεγαλύτερο μέρος του μενού αυτού το καταλαμβάνει μια σύνοψη του νέφους μας , δείχνοντας πληροφορίες για τις εικονικές μηχανές και τα εικονικά φυσικά χαρακτηριστικά που τις διέπουν καθώς και την καθολική δραστηριότητα εντός του οικοσυστήματος.



5.13 Το Περιβάλλον Nova και οι Hypervisors

Στο Nova περιβάλλον μας, που έχουμε ήδη αναλύσει, μπορούμε να δούμε τους Hypervisors που χρησιμοποιούμε. Στην περίπτωση μας, η υλοποίηση έγινε χρησιμοποιώντας τον **KVM** hypervisor, ο οποίος όπως παρατηρούμε στην παραπάνω εικόνα είναι σε λειτουργία. Ακόμα μπορούμε να δούμε στατιστικά στοιχεία σχετικά με τα instances που χρησιμοποιούνται, τα VCPUs, την χρήση της RAM, τις διαθέσιμες Floating IPs, τα Security Groups, τα Volumes και το Volume Storage.

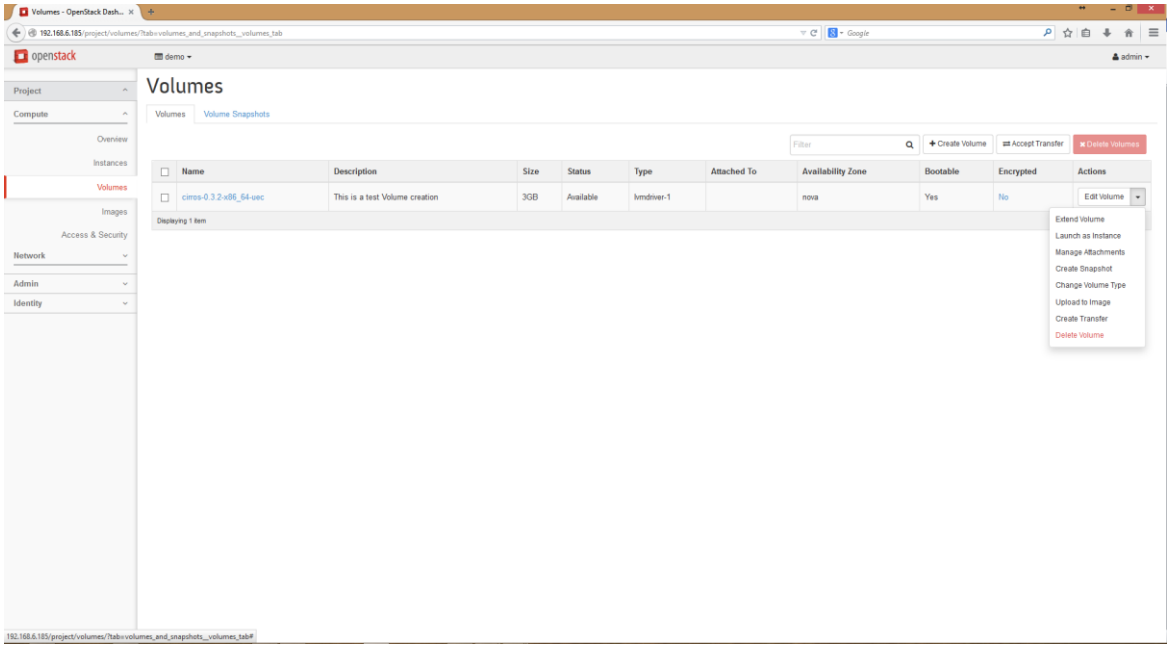


5.14 Τα Nova Instances που τρέχουν εντός του περιβάλλοντος

Περισσότερες πληροφορίες για την υπηρεσία Nova μπορούμε να δούμε στην καρτέλα Instances. Σε αυτήν μπορούμε να δούμε ποια Instances είναι σε λειτουργία και ενεργά καθώς και περαιτέρω πληροφορίες για αυτά όπως φαίνεται στην Εικόνα 5.14. Επίσης μας δίνεται η δυνατότητα για την διαχείριση των Instances μέσω περαιτέρω ενεργειών όπως:

- Associate Floating IP
- Disassociate Floating IP
- Edit Instance
- Edit Security Group
- Console
- View Log
- Pause Instance
- Suspend Instance

- Resize Instance
- Lock Instance
- Unlock Instance
- Soft Reboot Instance
- Hard Reboot Instance
- Shut Off Instance
- Rebuild Instance
- Terminate Instance



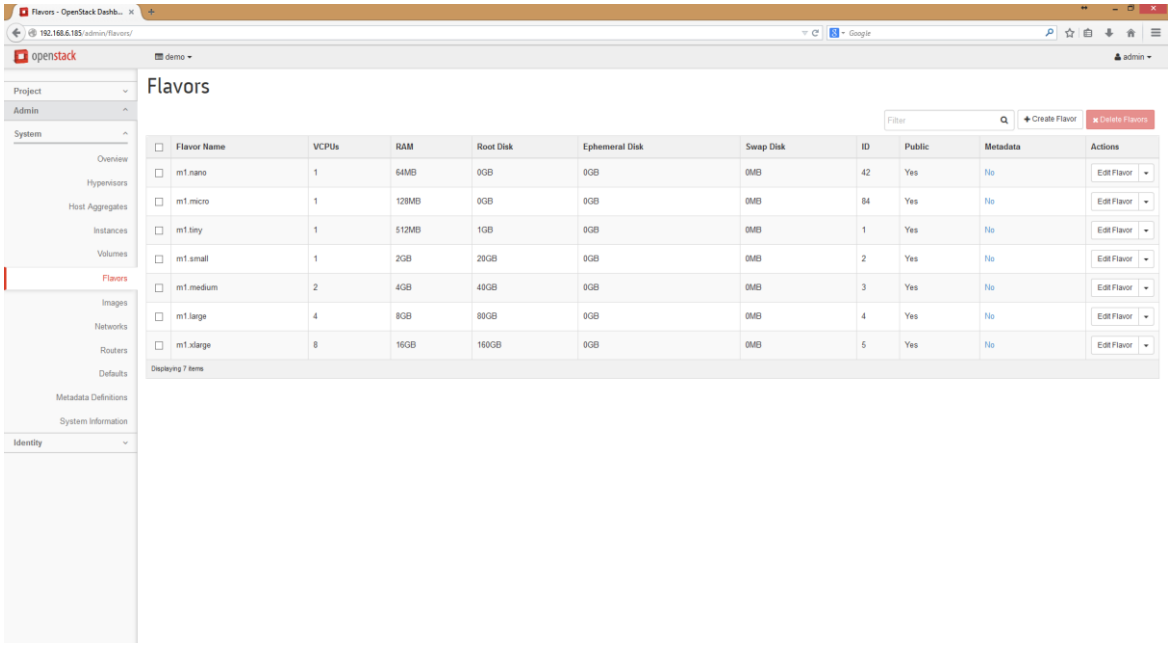
5.15 Η υπηρεσία Cinder

Όπως βλέπουμε στην εικόνα 5.15 , το περιεχόμενα της καρτέλας Volumes περιέχει πληροφορίες σχετικά με το Cinder και τους αποθηκευτικούς χώρους που φιλοξενούνται στο περιβάλλον μας. Για παράδειγμα στην εικόνα παραπάνω βλέπουμε ένα Volume με όνομα cirros-0.3.2-x86_64-uec. Στο πεδίο περιγραφής μπορούμε να προσθέσουμε ειδικές πληροφορίες σχετικά με το εκάστοτε Volume. Στην προκειμένη το Description συνοδεύεται από το σχόλιο “This is a test Volume creation”. Επίσης κατά την διαδικασία δημιουργίας μέσω του Wizard μας δίνεται η δυνατότητα επιλογής της χωρητικότητας που εδώ είναι 3GB. Επίσης υπάρχουν τα πεδία του Type, Attached to, που δηλώνει σε ποιο instance είναι προσκολλημένο το Volume, Availability Zone, Bootable (Yes/No) , Encrypted (Yes/No). Τέλος μέσω του μενού των περαιτέρω ενεργειών γίνονται διαθέσιμες οι παρακάτω ενέργειες:

- Extend Volume

- Launch an Instance
- Manage Attachments
- Create Snapshot
- Change Volume Type
- Upload to Image
- Create Transfer
- Delete Volume

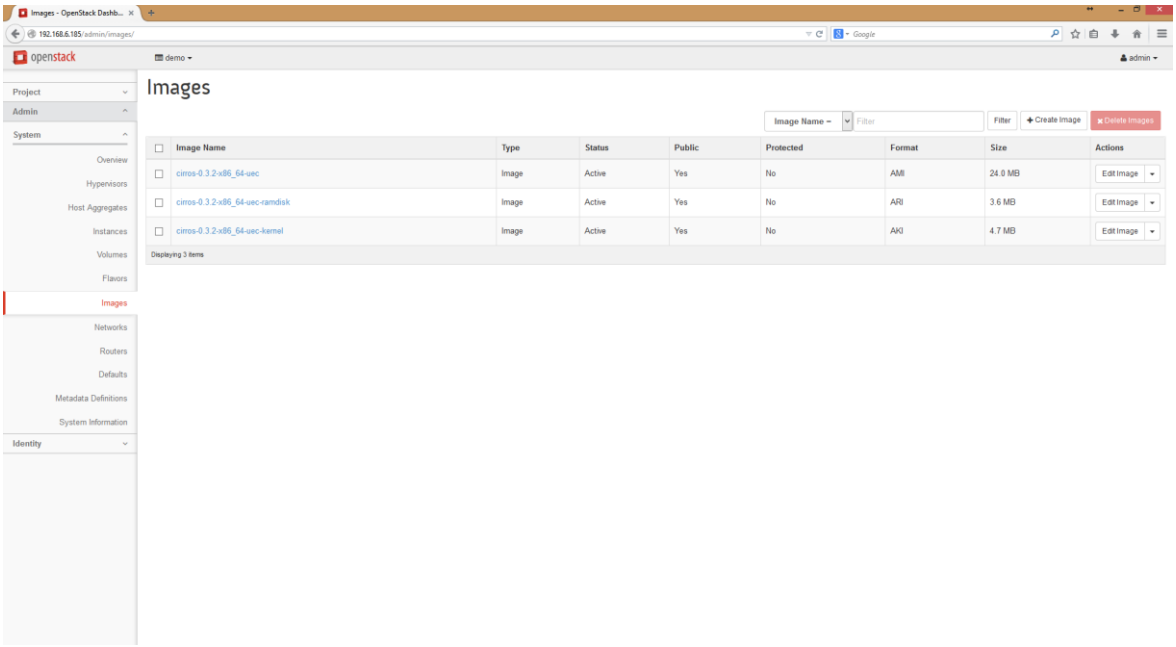
Από τις παραπάνω ενέργειες είναι πολύ σημαντική η επιλογή του **Upload to Image** και **Create Transfer**. Με την διαδικασία του **Upload to Image** ένας φοιτητής θα μπορεί να συνδέει το προσωπικό του Volume στο εκάστοτε συμβατό Image και να συνεχίζει την εργασία του εργαστηρίου. Επίσης με την δυνατότητα του **Create Transfer** δίνεται η δυνατότητα μεταφοράς του Volume με την παροχή ενός Transfer ID και του Authorization Key.



5.16 Flavors για τα προς δημιουργία Instances

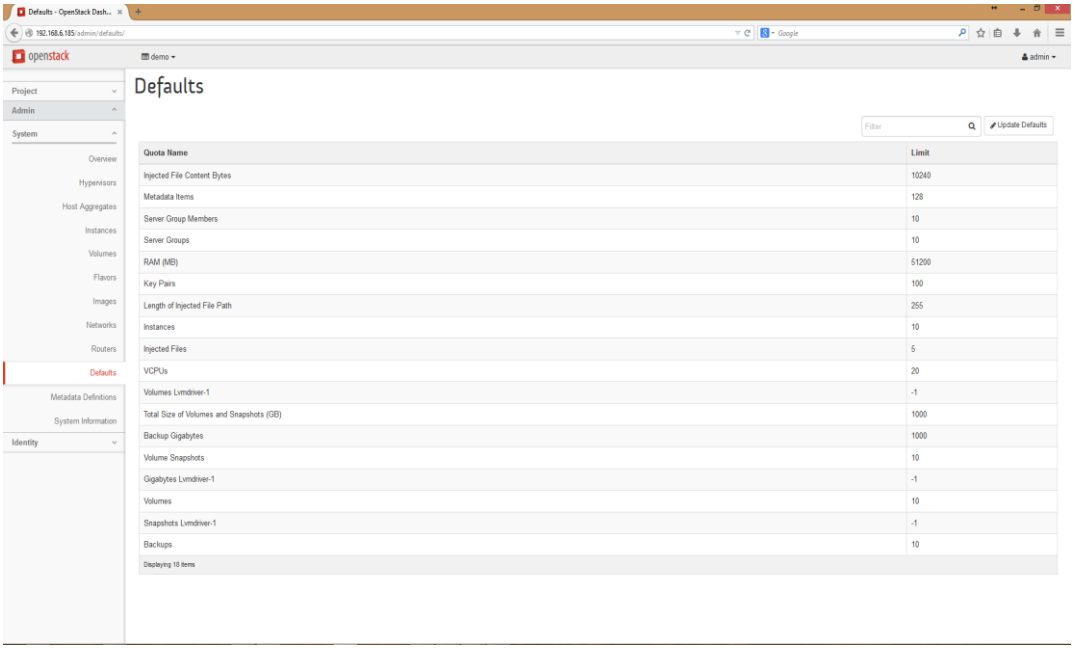
Στην εικόνα 5.16 βλέπουμε ρυθμίσεις οι οποίες ονομάζονται Flavors. Στην ουσία είναι προρυθμιζόμενες ιδιότητες για Instances που σκοπεύουμε να δημιουργήσουμε. Τέτοιες ρυθμίσεις όπως παρατηρούμε είναι ο αριθμός των εικονικών επεξεργαστών , η μνήμη , ο αποθηκευτικός χώρος κτλ. Επίσης μας δίνεται η δυνατότητα μέσω του Dashboard να μπορέσουμε να δημιουργήσουμε δικά μας ή να τροποποιήσουμε αυτά που έχουμε ήδη. Όλες αυτές οι επιλογές που μας δίνει το Dashboard μας γλυτώνουν από κόπο και χρόνο καθώς σε αντίθετη

περίπτωση θα έπρεπε να γίνουν μέσω εντολών python ή εργαλείων του orchestration API στο back-end περιβάλλον του Horizon.



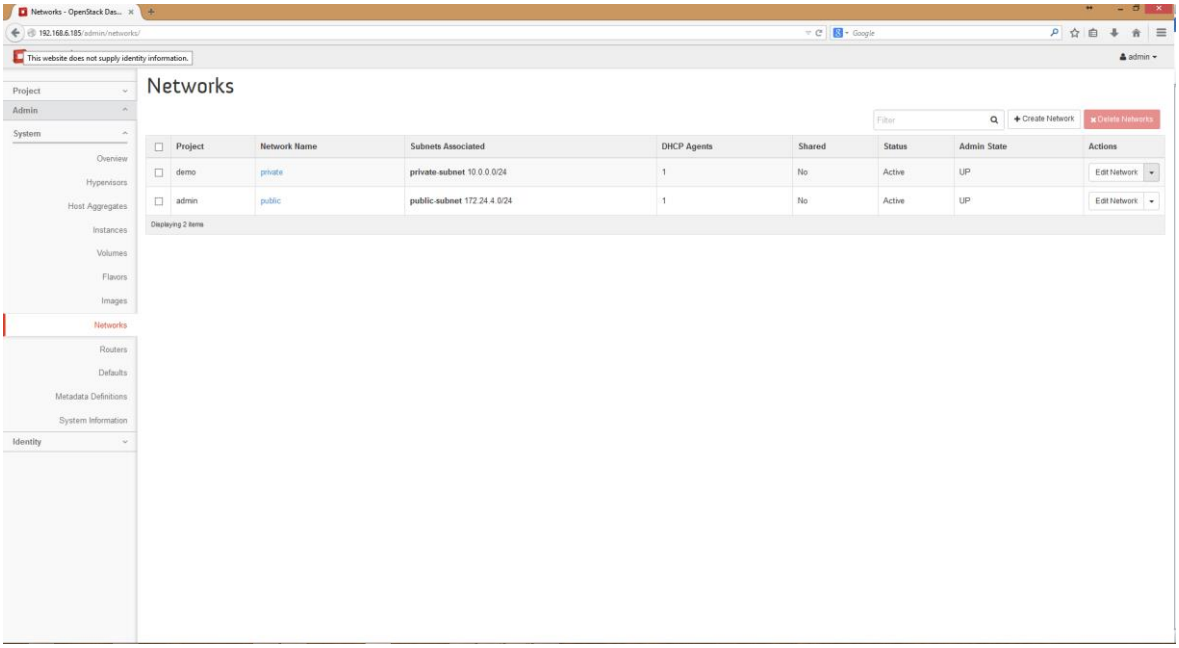
5.17 Η Υπηρεσία Glance και η διαθέσιμες εικόνες

Στην καρτέλα με τα Images το οποίο αποτελεί στην ουσία την διεπαφή του Glance μπορούμε να δούμε τα images τα οποία είναι διαθέσιμα προς χρήση. Στην καρτέλα αυτή οι επιλογές μας είναι συγκεκριμένες και είναι η δημιουργία και η διαγραφή κάποιας εικόνας.

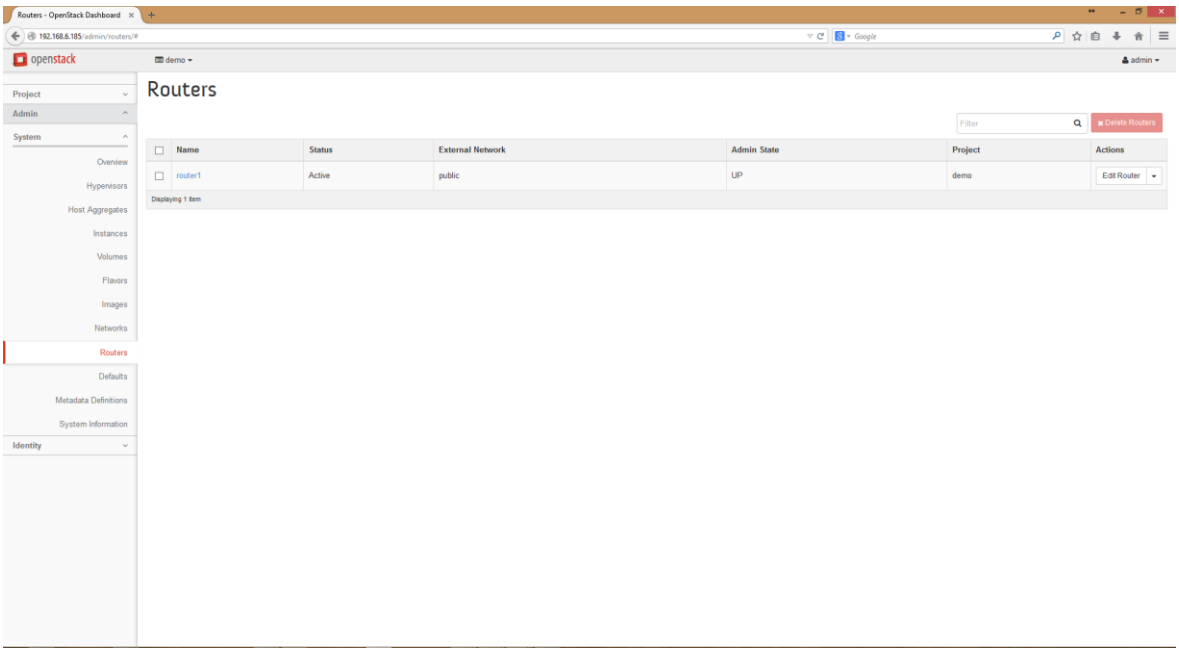


5.18 Οι προεπιλογές του συστήματος μας

Συνεχίζοντας με τις καρτέλες και τι μας προσφέρει η καθεμία από αυτές , έχουμε την καρτέλα Defaults. Η συγκεκριμένη καρτέλα μας δίνει σημαντικές πληροφορίες σχετικά με τα μέγιστα και ελάχιστα χαρακτηριστικά συστήματος που μπορούμε να χρησιμοποιήσουμε.



5.19 Τα δίκτυα του Περιβάλλοντος

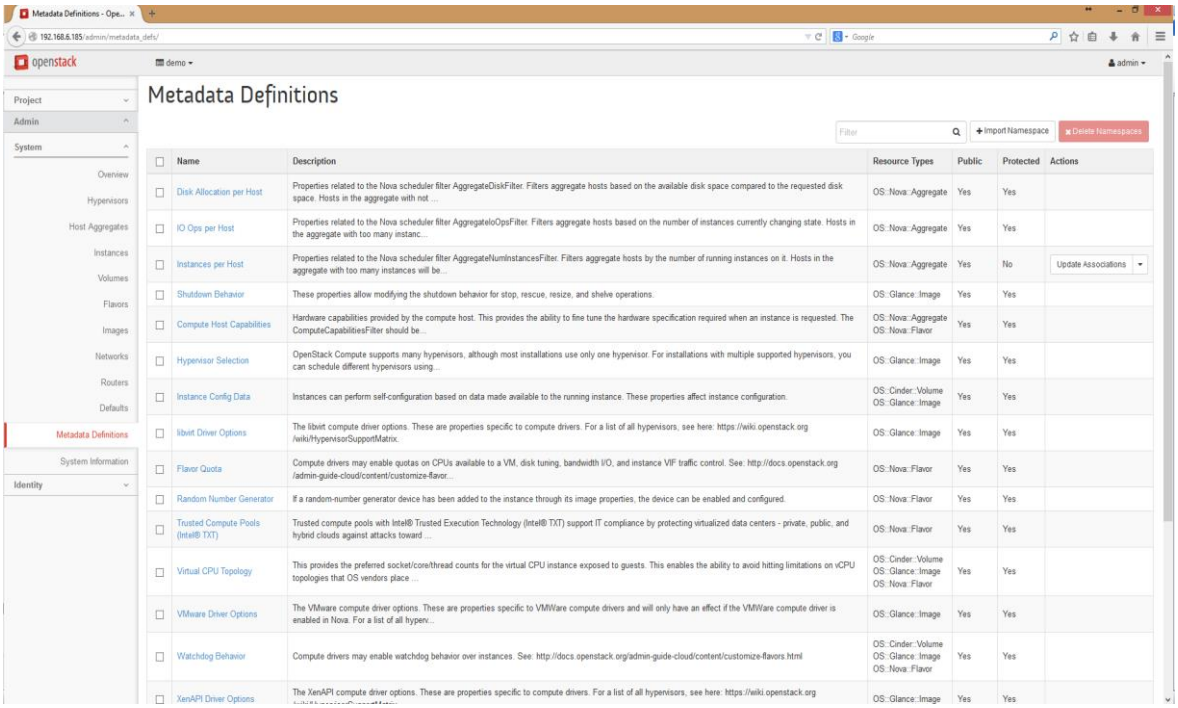


5.20 Οι δρομολογητές του Περιβάλλοντος

Στην παραπάνω 2 εικόνες βλέπουμε τις καρτέλες του έχουν να κάνουν με την υπηρεσία Neutron και το δικτυακό μέρος του περιβάλλοντος. Πιο συγκεκριμένα η εικόνα 5.17 μας παρουσιάζει πληροφορίες σχετικά με τα υπάρχοντα δίκτυα του

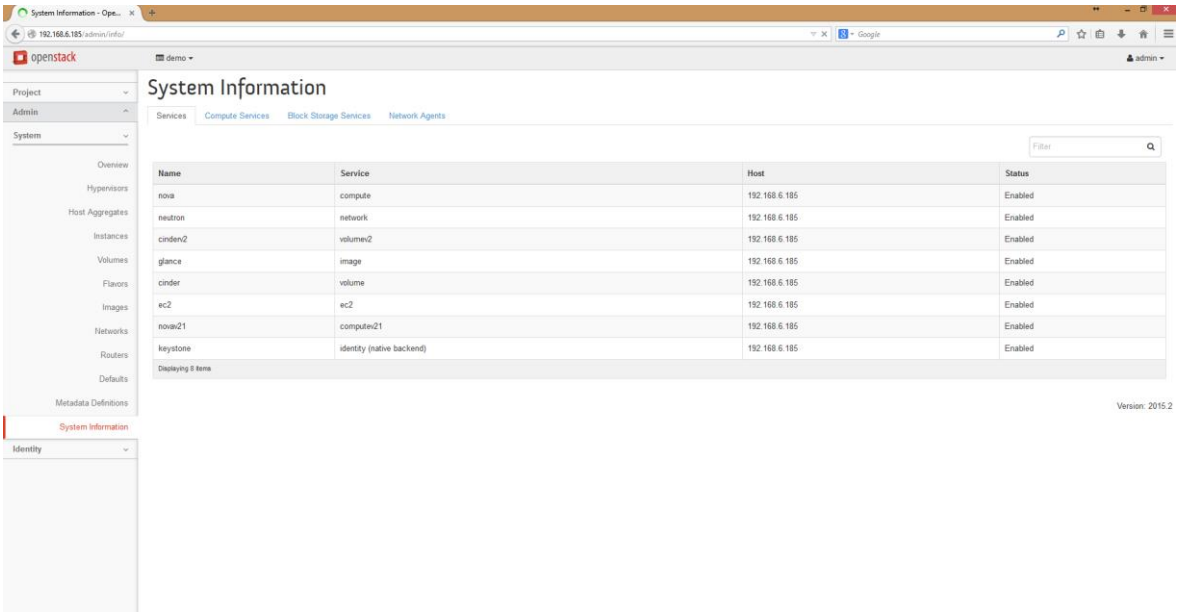
project. Όπως φαίνεται, υπάρχουν 2 δίκτυα, ένα ιδιωτικό για τα instances και ένα δημόσιο που λειτουργεί ως το μέσο για να δίνει εξωτερική πρόσβαση στο ιδιωτικό.

Στην συνέχεια στην εικόνα 5.18 φαίνονται οι δρομολογητές που υπάρχουν στο περιβάλλον μας με σκοπό να παρέχουν υπηρεσίες δρομολόγησης και σύνδεσης του περιβάλλοντος μας με το εξωτερικά δίκτυα και το Διαδίκτυο.



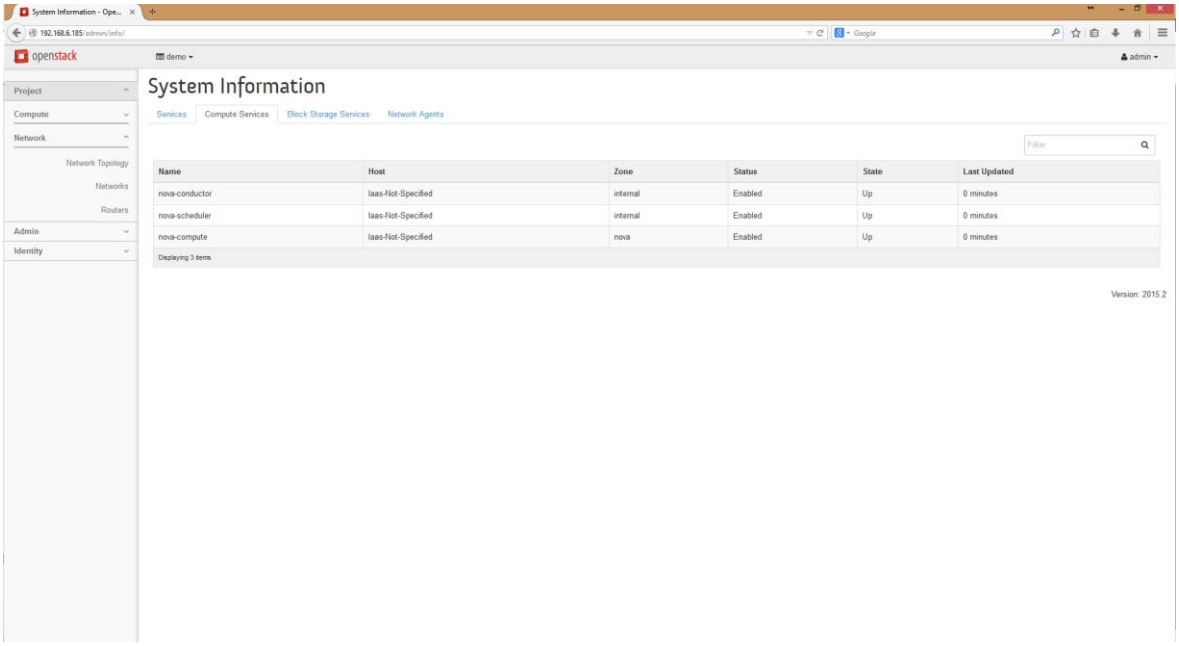
5.21 Οι ορισμοί των μεταδεδομένων

Στην καρτέλα Metadata Definitions υπάρχουν οι ορισμοί των μεταδεδομένων που υφίστανται εντός του περιβάλλοντος μας. Επίσης γίνεται ορατό από ποια υπηρεσία του Openstack προέρχονται καθώς και αν είναι ή δεν είναι Public και Protected.



5.22 Πληροφορίες συστήματος και Endpoints

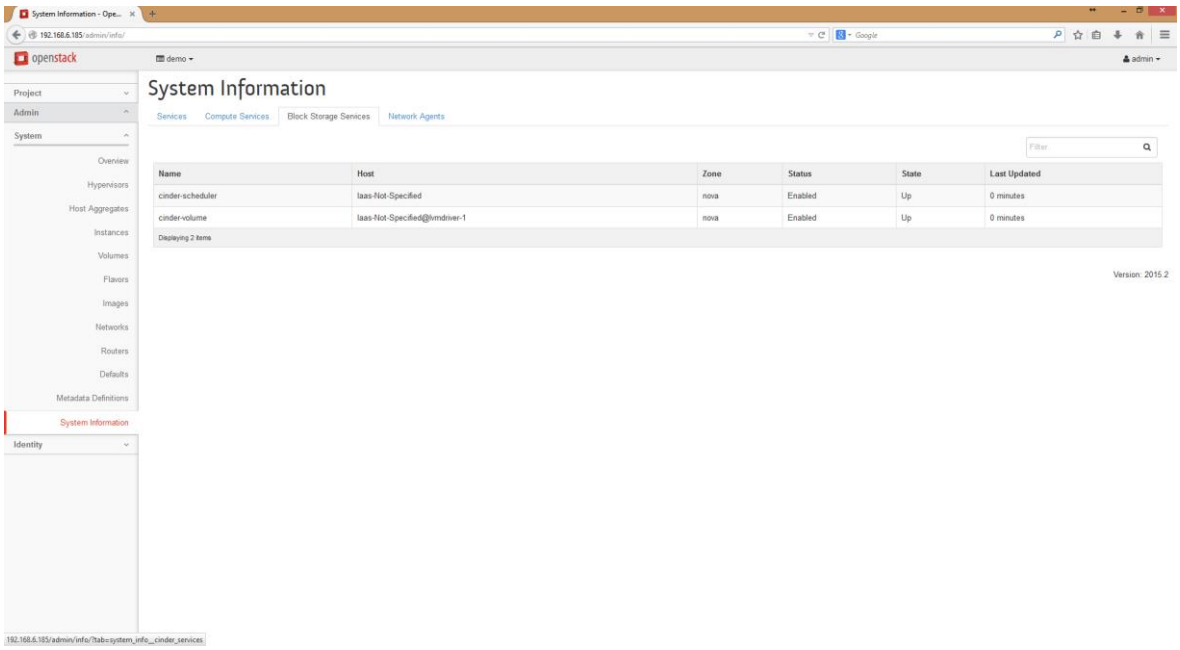
Στην καρτέλα System Info και πιο συγκεκριμένα στην καρτέλα Services που βρίσκεται στο εσωτερικό της, όπως φαίνεται και στην εικόνα 20 εμφανίζονται πληροφορίες συστήματος και τα διαφορετικά endpoints που έχουν δημιουργηθεί και οι υπηρεσίες που είναι διαθέσιμες. Επίσης φαίνονται και οι IPs που φιλοξενούν τις υπηρεσίες αυτές.



5.23 Πληροφορίες συστήματος και υπηρεσίες Nova

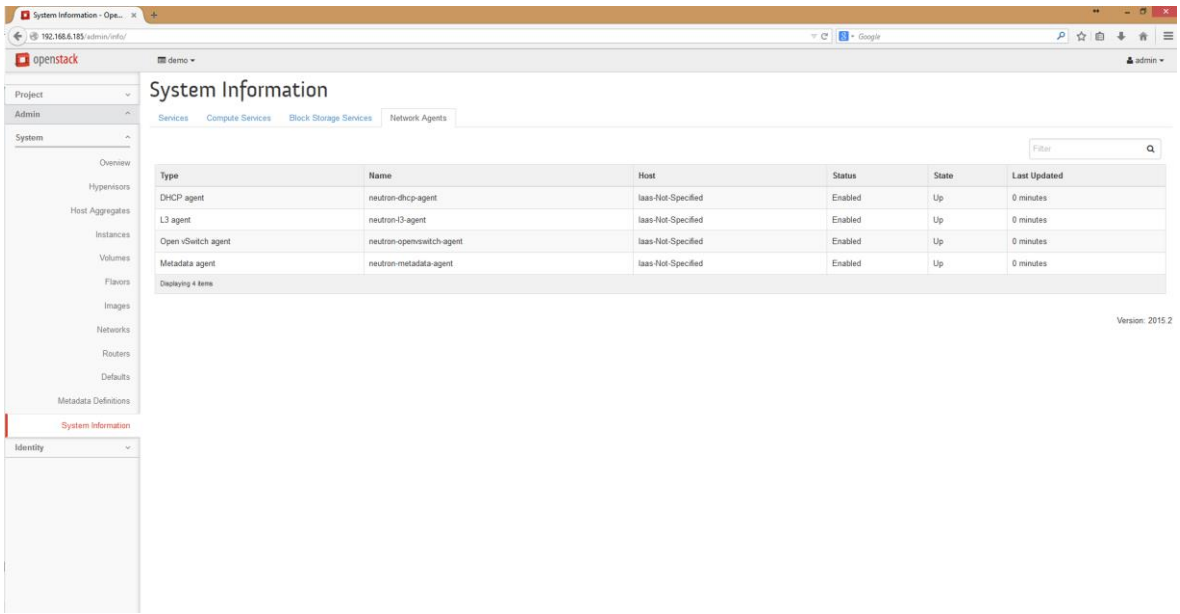
Στην αμέσως επόμενη καρτέλα Compute Services , υπάρχει μια λίστα με όλες τις υπηρεσίες του Nova που αποτελούν την επεξεργαστική δύναμη του

περιβάλλοντος μας .Επίσης δίνονται πληροφορίες σχετικά με την ζώνη που ανήκουν καθώς και ποια είναι η κατάσταση τους.



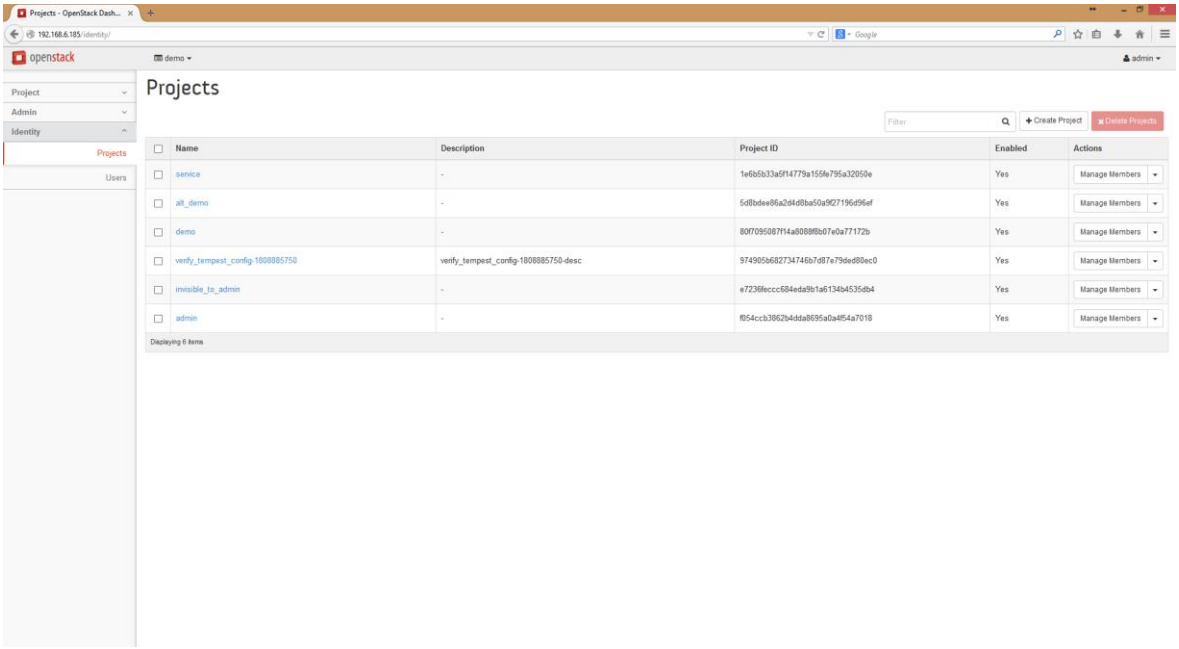
5.24 Πληροφορίες συστήματος και υπηρεσίες Cinder

Στην καρτέλα των Block Storages Services, μας δίνεται η δυνατότητα να δούμε ποιες είναι η ενεργές υπηρεσίες του Cinder. Χαρακτηριστικά, μπορούμε να δούμε ότι είναι οι ενεργές υπηρεσίες cinder-scheduler και cinder-volume. Επίσης, μας δίνονται πληροφορίες σχετικά με την Ζώνη στην οποία ανήκουν, την κατάσταση λειτουργίας του και τον πεπερασμένο χρόνο από την τελευταία ανανέωση.



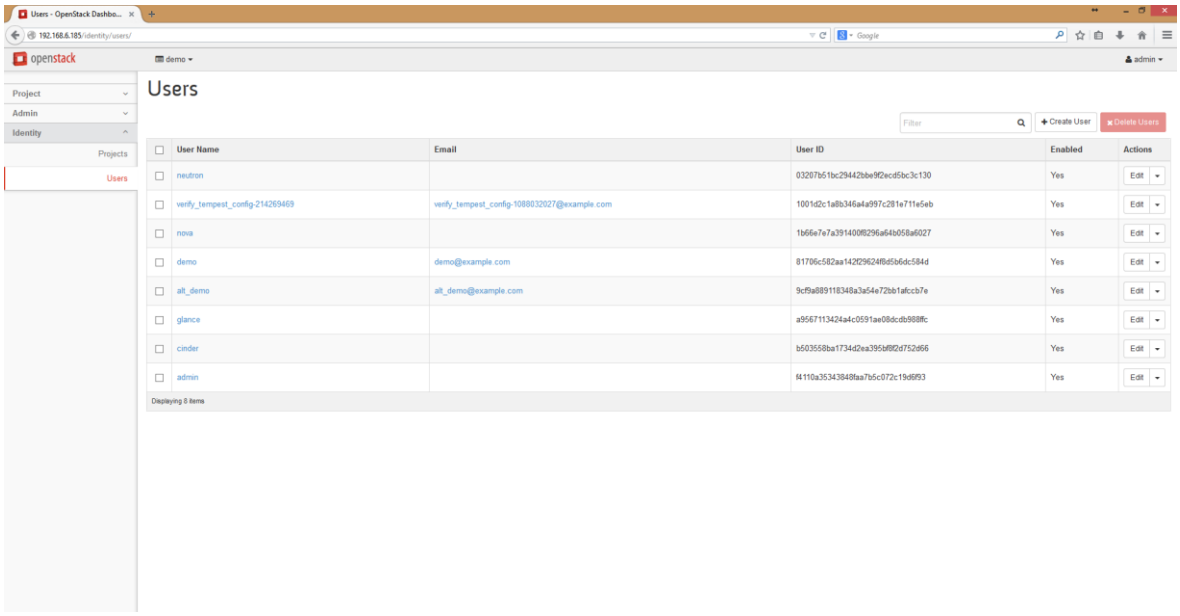
5.25 Πληροφορίες συστήματος και υπηρεσίες Neutron

Η τελευταία καρτέλα του System Information είναι αφιερωμένη στης υπηρεσίες του Neutron. Όπως μπορούμε να δούμε, οι υπηρεσίες που είναι ενεργές και λειτουργούν είναι οι DHCP agent, L3 agent, Open vSwitch agent και Metadata agent. Όπως και στις προηγούμενες καρτέλες, έτσι και εδώ, δίνονται πληροφορίες σχετικά με την κατάσταση λειτουργίας τον Host και τον χρόνο από την τελευταία ανανέωση.



5.26 Τα διαθέσιμα Projects

Στην παραπάνω καρτέλα φαίνονται τα projects μας, τα οποία είναι διαθέσιμα και ορισμένες πληροφορίες για το καθένα από αυτά. Η πληροφορία που φαίνεται είναι η τιμή στην στήλη Project ID. Το κάθε ID μεταφέρεται μέσω των APIs με προορισμό οποιαδήποτε υπηρεσία του Openstack . Εμείς μπορούμε να δούμε αυτά τα ID όταν δουλεύουμε στο εκάστοτε Project μέσω γραμμής εντολών.



5.27 Οι χρήστες

Όπως βλέπουμε στην κατηγορία των χρηστών υπάρχουν οι διαθέσιμοι χρήστες που υπάρχουν για το περιβάλλον μας. Εκτός από τον διαχειριστή που είναι προφανές ότι πρέπει να υπάρχει , δημιουργήσαμε και έναν χρήστη Demo. Αξίζει να παρατηρήσουμε ότι υπάρχει χρήστης για κάθε ξεχωριστή υπηρεσία που χρησιμοποιούμε. Αυτό συμβαίνει διότι , δεν πρέπει να ξεχνάμε πως κάθε υπηρεσίας έχει ένα λογαριασμό χρήστη εκχωρημένο στο περιβάλλον του Keystone που πρέπει να αυθεντικοποιείται και να ταυτοποιείται για να κάνει τις βασικές του λειτουργίες.

Μετά από την περιήγηση στο Dashboard σε αυτό το σημείο θα δούμε πως αναπτύχθηκε[49] αυτό.

Το πρώτο βήμα ήταν η λήψη των απαραίτητων πακέτων:

```
# apt-get install openstack-dashboard apache2 \
libapache2-mod-wsgi  memcached python-memcache
```

Αμέσως μετά για την ρύθμιση του dashboard έγιναν κάποιες αλλαγές και προσθήκες στο αρχείο /etc/openstack-dashboard/local_settings.py :

a. Ρυθμίστηκε το Dashboard έτσι ώστε να χρησιμοποιεί τις υπηρεσίες Openstack στον κόμβο ελέγχου:
    1. OPENSTACK_HOST = "controller"
b. Προστέθηκε η ρύθμιση έτσι ώστε οι hosts να έχουν πρόσβαση στο Dashboard:
    1. ALLOWED_HOSTS = ['*']

c. Ρυθμίστηκε η memcached συνεδρία της υπηρεσίας αποθήκευσης:

```
1.  CACHES = {
2.          'default': {
3.  'BACKEND':'django.core.cache.backends.memcached.
    MemcachedCache',
4.  'LOCATION': '127.0.0.1:11211',
5.          }
}
```

Επίλογος

Στο κεφάλαιο αυτό έγινε μια μακροσκελής ανάλυση του Openstack. Αρχικά είδαμε πως γεννήθηκε το Openstack και στην πορεία πως εξελίχθηκε ακολουθώντας και συμβαδίζοντας με τις τεχνολογικές εξελίξεις των καιρών. Στην συνέχεια, είδαμε που βασίστηκε η ανάπτυξη του δικού μας εγχειρήματος, **Lab as a Service,** ποιο λειτουργικό σύστημα, ποια τεχνολογία βάσεων δεδομένων , τι είδος και ποιοι hypervisors χρησιμοποιήθηκαν κτλ. Τέλος μελετήσαμε ξεχωριστά την καθεμία υπηρεσία που χρησιμοποιήθηκε για να είναι το Lab as a Service λειτουργικό. Παράλληλα, με την μελέτη είδαμε και πως αναπτύχθηκε και παραμετροποιήθηκε η κάθε υπηρεσία που ο συνδυασμός τους έχει σαν αποτέλεσμα την αρμονική λειτουργία του περιβάλλοντος μας.

6 Το Περιβάλλον υλοποίησης

Εισαγωγή

Σε αυτό τα κεφάλαιο θα αναφερθούμε στο περιβάλλον υλοποίησης του μοντέλου Lab as a Service. Θα αναλύσουμε την λειτουργικότητα του σε επίπεδο καθημερινής χρήσης και θα εμβαθύνουμε στον εξοπλισμό ο οποίος χρειάστηκε για την ανάπτυξη του. Αναφορές θα γίνουν τόσο στο υλικό (hardware) όσο και στο λογισμικό (software) που χρησιμοποιήθηκε και θα κλείσουμε το κεφάλαιο αυτό με μια μικρή ανάλυση γύρω από τον τρόπο στον οποίο λαμβάνει χώρα η αυθεντικοποίηση στο LaaS.

6.1 Εξοπλισμός του μοντέλου Lab as a Service

Για την εκπόνηση του συγκεκριμένου project είχε καταστεί εξ' αρχής γνωστό πως ήταν αναγκαία η ύπαρξη υλικού εξοπλισμού για την υποστήριξη ενός τέτοιου εγχειρήματος. Γι' αυτό το λόγο και μας δόθηκε από το A.TE.I με την συμβολή του κ. Ηλιούδη ένας server πάνω στον οποίο στήθηκε όλο το περιβάλλον του υλοποίησης μας.

6.1.1 Hardware

Πρόκειται για το «System x3650 M2» μοντέλο-διακομιστή της IBM με τα ακόλουθα χαρακτηριστικά:

- **Επεξεργαστής:** 8 x Intel(R) Xeon(TM) CPU 3.20GHz
- **Μνήμη RAM:** 5079MB (4620MB χρησιμοποιούνται)
- **Σκληρός δίσκος:** SATA Hard Drive 899.7GB
- **Κάρτα γραφικών:** Advanced Micro Devices, Inc. [AMD/ATI] ES1000er])

Το κομμάτι hardware που μας δόθηκε κατασκευάζεται και συντίθεται από την IBM και έχει μια πληθώρα διαφορετικών εκδόσεων με τα δικά της χαρακτηριστικά η κάθε μία. Τα χαρακτηριστικά τα οποία πληροί το δικό μας project είναι μια μέση λύση για μια πιλοτική υλοποίηση και θα αναφερθούμε εκτενέστερα για το πόσο φόρτο μπορεί να υποστηρίξει και ποιες είναι οι δυνητικές προϋποθέσεις επέκτασης. Αξίζει να αναφερθεί πως μια πλήρης λίστα

χαρακτηριστικών του μοντέλου που χρησιμοποιούμε μπορεί να βρεθεί στο επίσημο δικτυακό ιστότοπο της IBM (http://www-01.ibm.com/common/ssi/rep_ca/1/877/ENUSZG09-0661/ENUSZG09-0661.PDF) όπου μπορούμε να δούμε αναλυτικά τι δυνατότητες μπορεί να επιτύχει το συγκεκριμένο μοντέλο.

Έμφαση αξίζει να δοθεί από το παραπάνω σύνδεσμο στο ότι ο αριθμός των επεξεργαστών το συγκεκριμένο μοντέλο μπορεί να αυξηθεί στους 12 ενώ η διαθέσιμη προσωρινή μνήμη RAM μπορεί να φτάσει τα 128GB χρησιμοποιώντας modules των 8GB. Θα αναλύσουμε αργότερα τις απαιτήσεις τις υλοποίησης μας και το κατά πόσο μια αναβάθμιση προτείνεται προκειμένου να υποστηριχθεί ο φόρτος ολόκληρου του ιδρύματος.

Όμοια με την επεξεργαστική ισχύ και την προσωρινή μνήμη το συγκεκριμένο μοντέλο προσφέρει εξαιρετική δυνατότητα επέκτασης του αποθηκευτικού χώρου. Από τα 900GB που έχουμε διαθέσιμα αυτή τη στιγμή μπορούμε να προσθέσουμε 4 έως 8 σκληρούς δίσκους τύπου SAS (ειδικοί δίσκοι μακράς διάρκειας για διακομιστές) και να εκτοξεύσουμε την χωρητικότητα σε μεγάλα νούμερα κάτι το οποίο θα αποδειχθεί ιδιαίτερα χρήσιμο αναλόγως του τι χωρητικότητα θέλουμε να αφιερώσουμε σε κάθε χρήστη του νέφους μας.

Τέλος ο επεξεργαστής γραφικών (κάρτα γραφικών) δεν επιδέχεται κάποιας αναβάθμισης ή επέκτασης αλλά πρόκειται για χαρακτηριστικό ελάσσονος σημασίας για το συγκεκριμένο εγχείρημα καθώς ο συγκεκριμένος επεξεργαστής δεν επιβαρύνεται καθόλου αφού δεν υπάρχουν γραφικά τα οποία πρέπει να μεταδοθούν μέσω του νέφους σε απαιτητικό βαθμό.

6.1.2 Software

Το software όπως γνωρίζουμε και από τα προηγούμενα κεφάλαια, χωρίζεται στο λειτουργικό που φιλοξενεί το μοντέλο μας, τον hypervisor και την πλατφόρμα η οποία υλοποιεί το νέφος μας.

Ως λειτουργικό πάνω στον server χρησιμοποιήσαμε Linux Ubuntu Server, την έκδοση 14.02 και όπως έχουμε ήδη δει στο προηγούμενο κεφάλαιο ενεργοποιήσαμε τον ενσωματωμένο KVM hypervisor που βρίσκεται στον kernel του λειτουργικού. Για την υλοποίηση του νέφους βασιστήκαμε στην πλατφόρμα

Openstack στην τελευταία και πιο πρόσφατη έκδοση της με κωδική ονομασία “Juno”.

6.2 Οι Clients

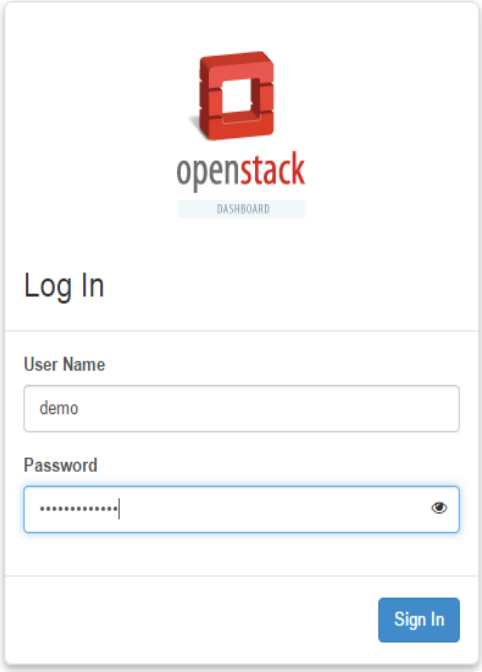
Όπως έχει σημειωθεί και προηγουμένως δυνητικά οι clients του συγκεκριμένου συστήματος είναι οποιοσδήποτε ηλεκτρονικός υπολογιστής έχει πρόσβαση στο εσωτερικό δίκτυο του ΤΕΙ μέσω της υπηρεσίας VPN που παρέχεται. Αξίζει να σημειωθεί σε αυτό το σημείο ότι δεν υπάρχουν συγκεκριμένες απαιτήσεις (υλικού ή λογισμικού) από τους clients. Αυτό συμβαίνει για δυο κυρίως λόγους. Στο μοντέλο μας η τελική πρόσβαση του χρήστη επιτυγχάνεται μέσω web interface. Αυτό σε πρακτικό επίπεδο σημαίνει ότι οποιοσδήποτε χρήστης επιθυμεί να εισέλθει στις υπηρεσίες που παρέχονται από το νέφος μας αρκεί να έχει έναν από τους πρόσφατους browser που έρχονται είτε ενσωματωμένοι στο λειτουργικό είτε μπορεί να κατεβάσει δωρεάν. Επίσης το Openstack δεν υποστηρίζει λειτουργία thin-fat clients. Συνεπαγωγικά λοιπόν, κατανοούμε πως όλο τον φόρτο των διεργασιών τον επωμίζεται ο διακομιστής του Νέφους και όχι ο τοπικός υπολογιστής μέσω του οποίου εισερχόμαστε στην υπηρεσία.

6.3 Χρήση του συστήματος ως απλός χρήστης (Web-Interface)

6.3.1 Είσοδος στο σύστημα

Για την είσοδο στο σύστημα του νέφους θα χρειαστούμε έναν οποιονδήποτε browser. Ακολούθως βεβαιωνόμαστε πως είμαστε συνδεδεμένοι στο ίντερνετ και έχουμε IP που ανήκει στο εσωτερικό δίκτυο του ΑΤΕΙ χρησιμοποιώντας την υπηρεσία VPN που παρέχεται από το τμήμα πληροφορικής.

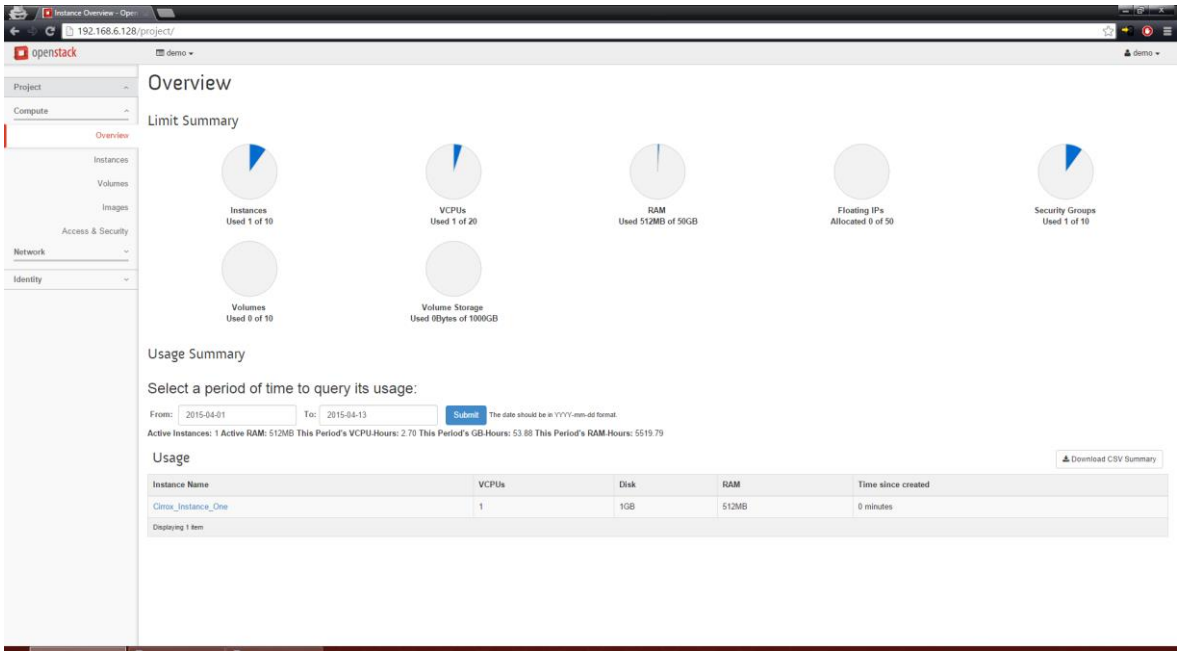
Πληκτρολογούμε την διεύθυνση σύνδεσης η οποία είναι 192.168.6.185 και όπως είδαμε ήδη από το κεφάλαιο 5 η βασική front-end διεπαφή είναι η παρακάτω.



6.1 Η διεπαφή εισόδου του χρήστη

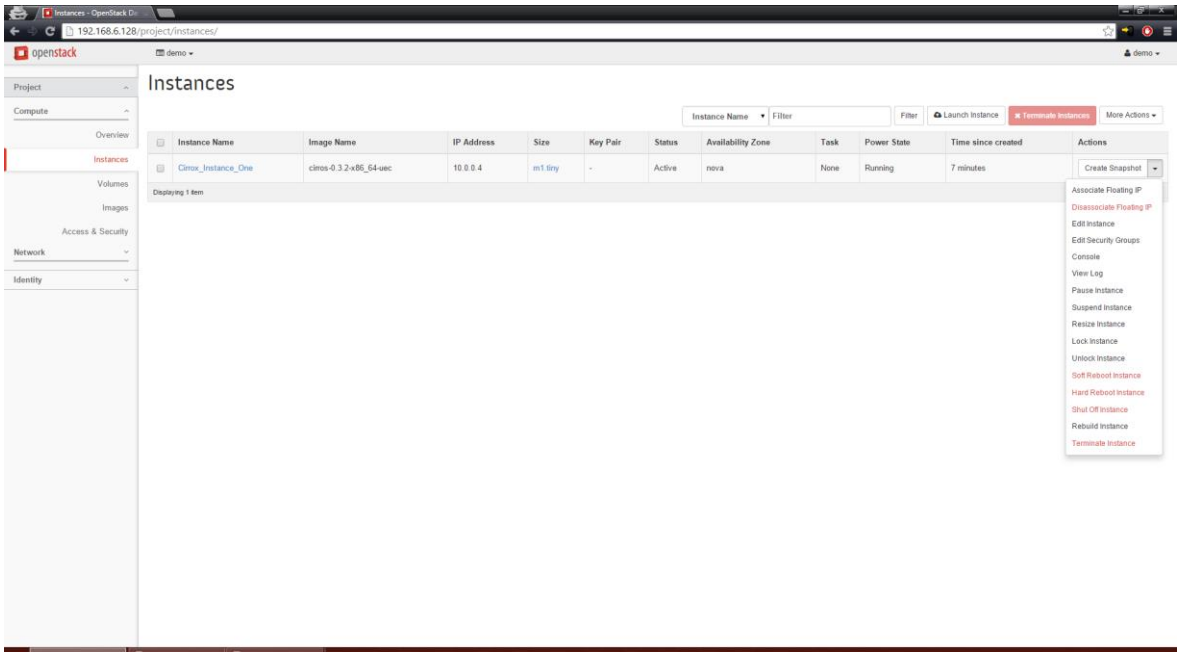
Ως απλός χρήστης θα χρησιμοποιήσουμε το όνομα χρήστη και τον κωδικό ο οποίος μας δόθηκε για την είσοδο μας στο σύστημα. Εφόσον η είσοδος μας είναι επιτυχής κάτι το οποίο θα ελεγχθεί μέσω του συστήματος αυθεντικοποίησης θα μεταβούμε σε διαφορετικό interface.

6.3.2 Ανάλυση δυνατοτήτων διεπαφής χρήστη



6.2 Η Κεντρική Προεπισκόπηση για τον χρήστη

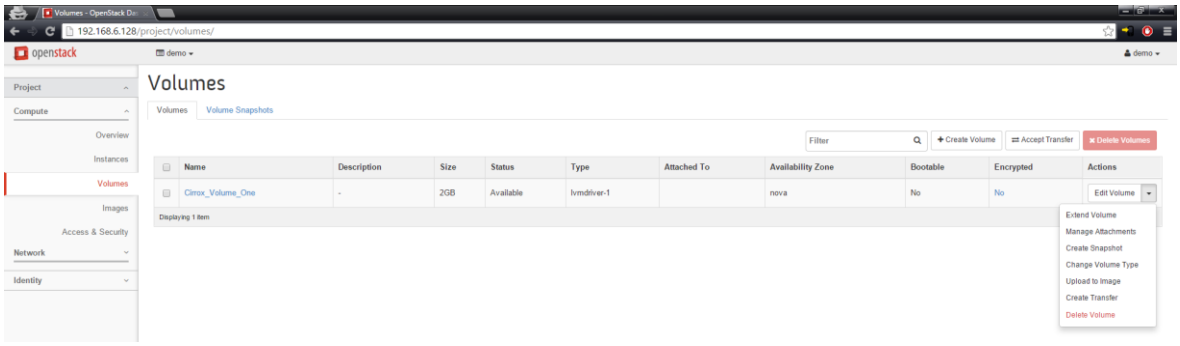
Όπως και ο διαχειριστής έτσι και ο απλός χρήστης στην πρώτη καρτέλα «Overview» μπορεί να δει συγκεντρωτικά πληροφορίες που αφορούν τη διαχείριση του λογαριασμού του μέσα στο συνολικό project. Με πιο απλοϊκά λόγια μπορεί να δει τι «στοιχίζει» ο ίδιος σε φόρτο στον κεντρικό διακομιστή του Openstack και να λάβει τις κατάλληλες αποφάσεις για το πώς θα συνεχίσει την εργασία του και τι χρειάζεται να διατηρήσει ενεργό από τα εικονικά του μηχανήματα. Για την βοήθεια του σε αυτές τις αποφάσεις η κεντρική σελίδα περιέχει δείκτες για των αριθμό των instances που τρέχουν μια δεδομένη στιγμή, καθώς και πληροφορίες επεξεργαστικής ισχύος, κατανάλωσης προσωρινής μνήμης RAM, δέσμευση διαθέσιμων εικονικών διευθύνσεων IP, διαθέσιμων ομάδων ασφαλείας και τέλος πληροφορίες χωρητικότητας που καταλαμβάνεται. Αξίζει να σημειωθεί πως ο χρήστης σε αντίθεση με τον διαχειριστή βλέπει αποκλειστικά και μόνο αυτά που αφορούν τον δικό του λογαριασμό και όχι το σύνολο – όπως εξάλλου είναι ευνόητο ότι θα πρέπει να συμβαίνει.



6.3 Το μενού Instances του χρήστη

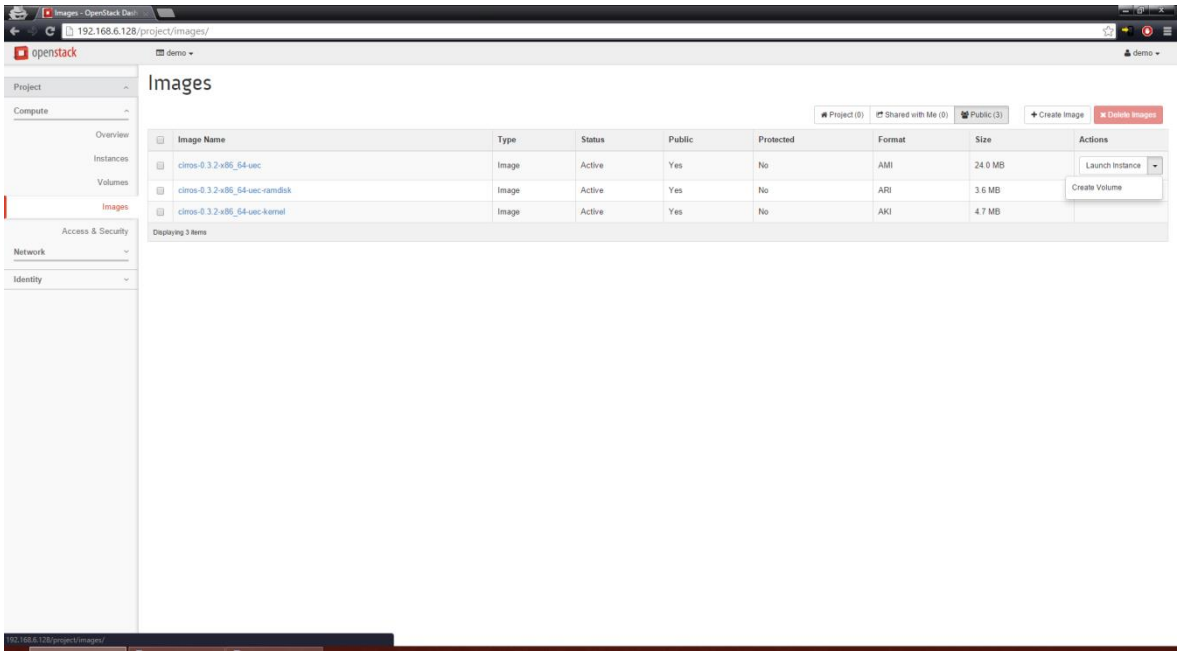
Προχωρώντας στην επόμενη καρτέλα «Instances» κατά παρόμοιο τρόπο με το προηγούμενο κεφάλαιο, ο χρήστης μπορεί να δει για τα ήδη τρέχοντα instances συγκεντρωτικές πληροφορίες σε ότι αφορά το λειτουργικό, την διεύθυνση IP, το μέγεθος τους, την κατάσταση καθώς και για πόσο διάστημα είναι αυτά ενεργά. Πάνω σε κάθε instance δίνεται στον χρήστη η δυνατότητα κάποιων ενεργειών.

Οι ενέργειες αυτές χωρίζονται σε ενέργειες που αφορούν την επεξεργασία του instance σαν οντότητα (έτσι ο χρήστης έχει την δυνατότητα να επεξεργαστεί τα χαρακτηριστικά του, να το αναδιαμορφώσει, να το παύσει, να το κλειδώσει/ξεκλειδώσει ή να το τερματίσει), αφορούν το λειτουργικό το οποίο εκτελείται στο instance και άρα δίδονται στον χρήστη δυνατότητες soft/hard reboot αλλά και shut down ή αφορούν την δικτύωση του συγκεκριμένου instance (έτσι μπορεί ο χρήστης να δώσει μια συγκεκριμένη εικονική IP στο instance – από το pool των διαθέσιμων IP που έχει ορίσει ο διαχειριστής) αν και η διαδικασία αυτή στην δική μας περίπτωση είναι αυτοματοποιημένη.



6.4 Το μενού Volumes του χρήστη

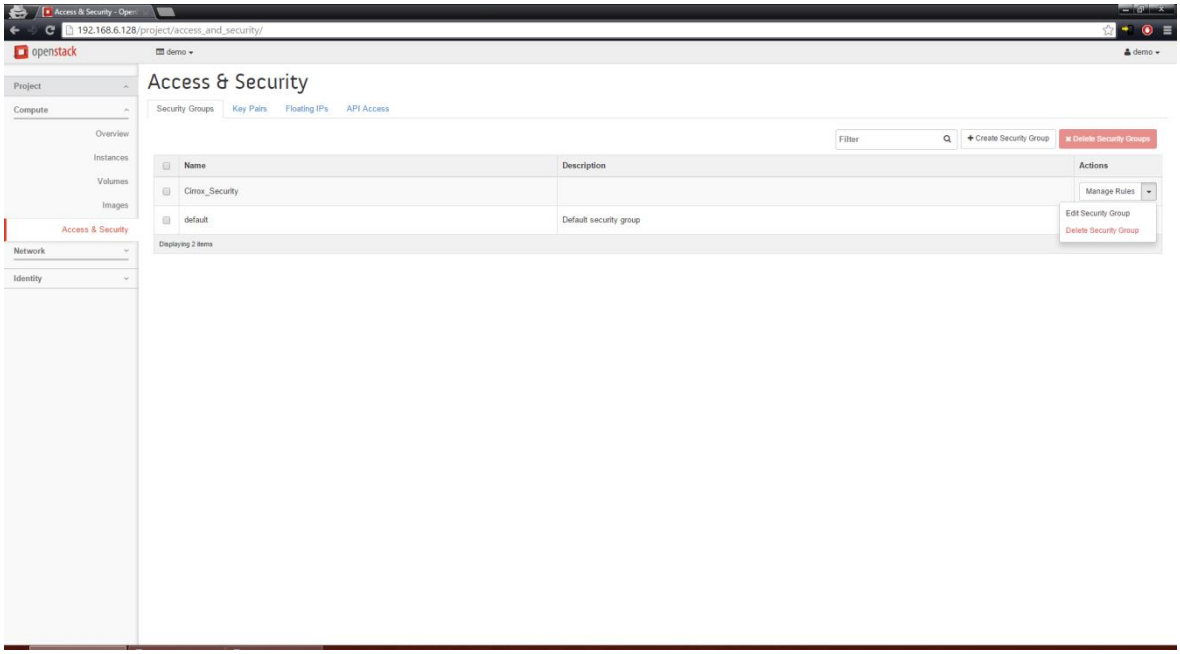
Στην ακόλουθη καρτέλα «Volumes» ο χρήστης μπορεί να δημιουργήσει αποθηκευτικό χώρο ο οποίος θα δίνεται σε συγκεκριμένα Instances (προσομοιώνοντας έτσι την φιλοσοφία ενός σκληρού δίσκου λειτουργικού συστήματος) είτε θα διαχωρίζεται από αυτά ανάλογα τις ανάγκες μας. Ο χρήστης έχει ανά πάσα στιγμή επιλογές δυναμικής αυξομείωσης του αποθηκευτικού χώρου, αλλαγής του τύπου του ή/και διαγραφής του καθώς και προσκόλλησης ή αποκόλλησης του σε κάποια συγκεκριμένη εικονική μηχανή ή instance.



6.5 Το μενού Images του χρήστη

Συνεχίζοντας την πλοήγηση μας στις διαθέσιμες στον χρήστη καρτέλες, συναντούμε την καρτέλα «Images». Η καρτέλα αυτή αποτελεί την διεπαφή που μας επιτρέπει στον χρήστη να δει όλες τις εικόνες των λειτουργικών συστημάτων που μπορεί να του παρέχει το νέφος μας και που είναι έτοιμες προς εκκίνηση και

χρήση. Από ένα image σε αυτή την καρτέλα, ο χρήστης μπορεί (προαιρετικά) να δημιουργήσει έναν αποθηκευτικό χώρο και να εκκινήσει ένα instance κάποιου λειτουργικού.

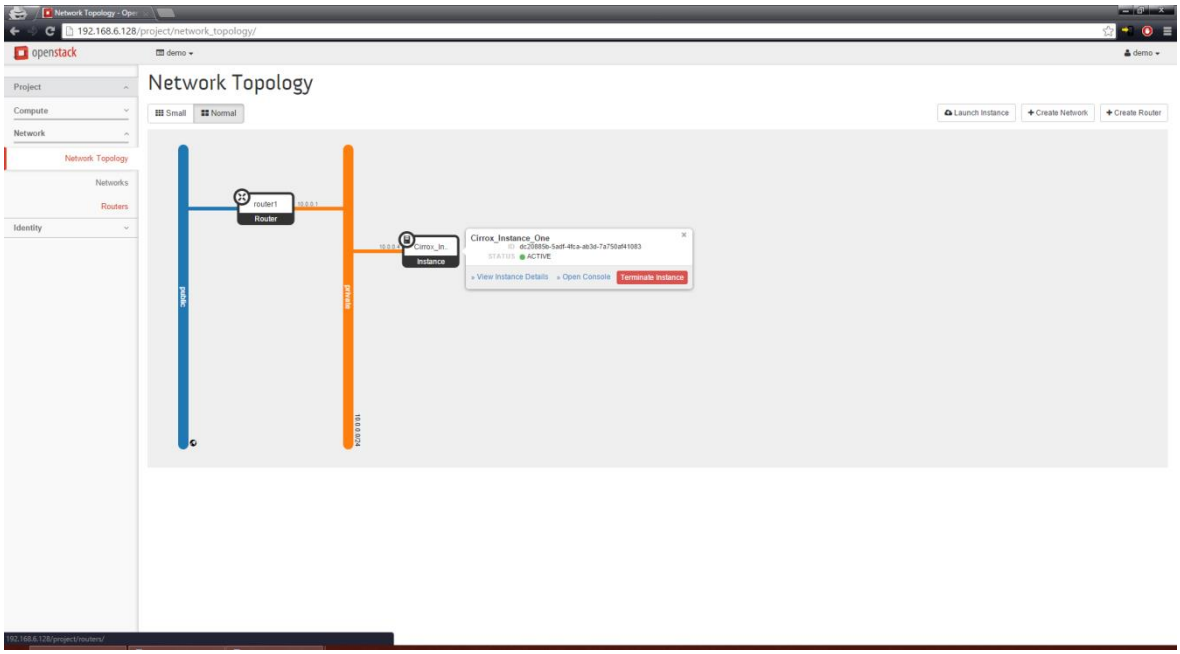


6.6 Το μενού Access & Security του χρήστη

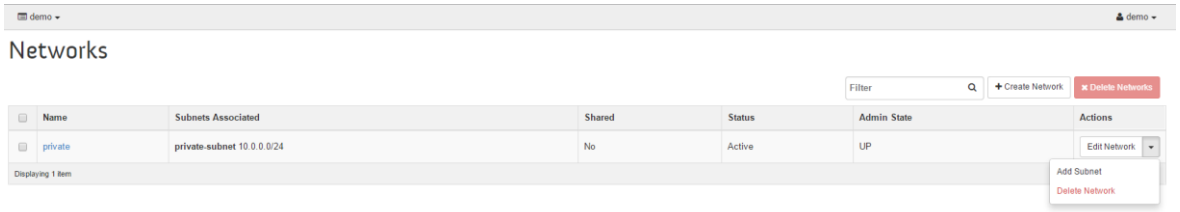
Τελευταία καρτέλα για το «Compute» σε περιβάλλον χρήστη αποτελεί το «Access & Security». Εδώ ο χρήστης στα πλαίσια που του επιτρέπει ο διαχειριστής μπορεί να δημιουργήσει ομάδες ασφαλείας. Κάθε ομάδα ασφαλείας διέπεται από κανόνες οι οποίοι με την σειρά τους αφορούν στην κίνηση (traffic) των δεδομένων εντός και εκτός των εικονικών λειτουργικών συστημάτων.

Έτσι ο χρήστης μπορεί να θέσει κανόνες για κάθε instance (εφόσον το εντάξει στην κατάλληλη ομάδα ασφαλείας) σχετικά με τον τύπο διεπαφής, το πρωτόκολλο διευθυνσιοδότησης, το εύρος των χρησιμοποιούμενων πορτών κ.ά. Σε ιδανικές συνθήκες για απροβλημάτιστη λειτουργία καλό θα ήταν ο χρήστης να ακολουθεί τα προρυθμισμένα (default) γκρουπ και κανόνες ασφαλείας όπως έχουν οριστεί από τον διαχειριστή και να μην επεμβαίνει σε αυτά.

Αναλυτικότερα για το πώς μπορεί ο χρήστης να δημιουργήσει instance, αποθηκευτικό χώρο ή instance από την διεπαφή των images, ομάδες ασφαλείας και σετ κανόνων ασφαλείας θα δούμε παρακάτω και αφού ολοκληρώσουμε την περιήγηση μας σε όλη τη διεπαφή του dashboard.



6.7 Το μενού με την τοπολογία του δικτύου



6.8 Το μενού με τα δίκτυα του περιβάλλοντος

6.9 Το μενού με δρομολογητές του περιβάλλοντος

Στο επόμενο μενού με την γενική ονομασία «Networks» ο χρήστης μπορεί εν συντομία να επεξεργαστεί οτιδήποτε αφορά στο εσωτερικό εικονικό δίκτυο στο οποίο εντάσσονται οι εικονικές μηχανές οι οποίες δημιουργούνται στο νέφος μας. Πιο αναλυτικά, στο:

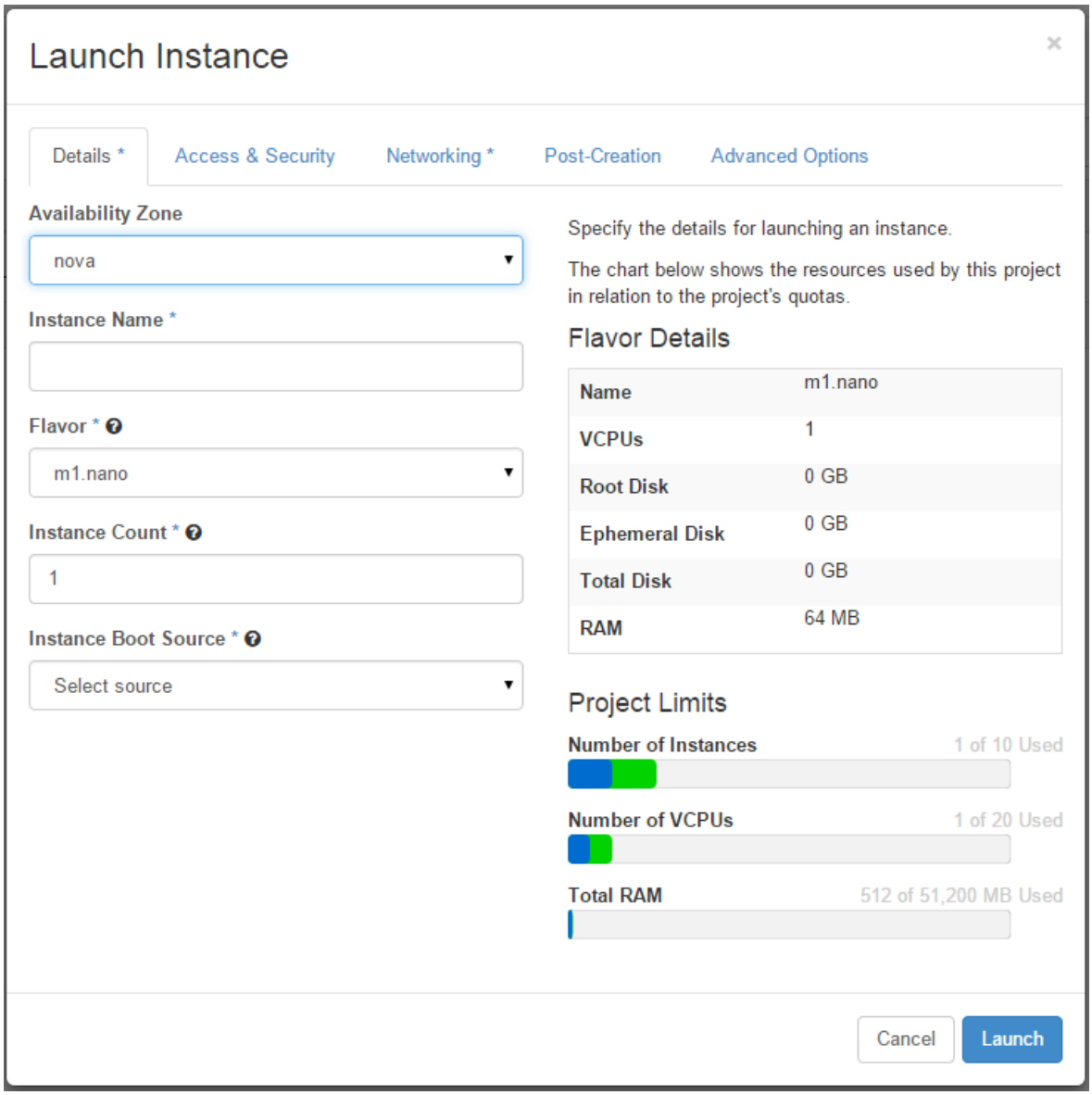
- **Network Topology:** Εδώ μπορεί ο χρήστης να δει σχηματικά μια αναπαράσταση της τοπολογίας δικτύου έτσι όπως αυτή αναπτύσσεται βάσει των δημιουργηθέντων instances.
- **Networks:** Εδώ δίνεται η δυνατότητα για επεξεργασία (ή δημιουργία νέων) εικονικών δικτύων και υποδικτύων στα οποία μπορούν να εντάσσονται τα εικονικά μηχανήματα.
- **Router:** Εδώ γίνεται η δυνατότητα επεξεργασίας του εικονικού αντικειμένου που λειτουργεί ως router μέσα στο virtual network στο οποίο είναι ενταγμένο το εικονικό μας μηχάνημα.

6.3.3 Δημιουργία Instances και αποθηκευτικού χώρου

6.3.3.1 Δημιουργία ενός νέου instance

Μέσα στο νέφος δίνεται η δυνατότητα στον χρήστη να ξεκινήσει πολλαπλά instances ανάλογα με τις ανάγκες της φύσης της δουλειάς του και τους διαθέσιμους πόρους που θα του παρέχει το νέφος.

Από το μενού «Compute» στην καρτέλα «Instances» αρκεί να πατήσει το κουμπί «Launch Instance». Αυτή η ενέργεια θα φέρει στο προσκήνιο την παρακάτω διεπαφή για την δημιουργία ενός νέου εικονικού μηχανήματος το οποίο θα τρέχει το instance του εικονικού λειτουργικού που θα επιλέξουμε.



6. Επιλογές κατά την δημιουργία Image

Η απαραίτητη καρτέλα την οποία υποχρεούται να συμπληρώσει ο χρήστης σε αυτό το νέο παράθυρο είναι η καρτέλα «Details».

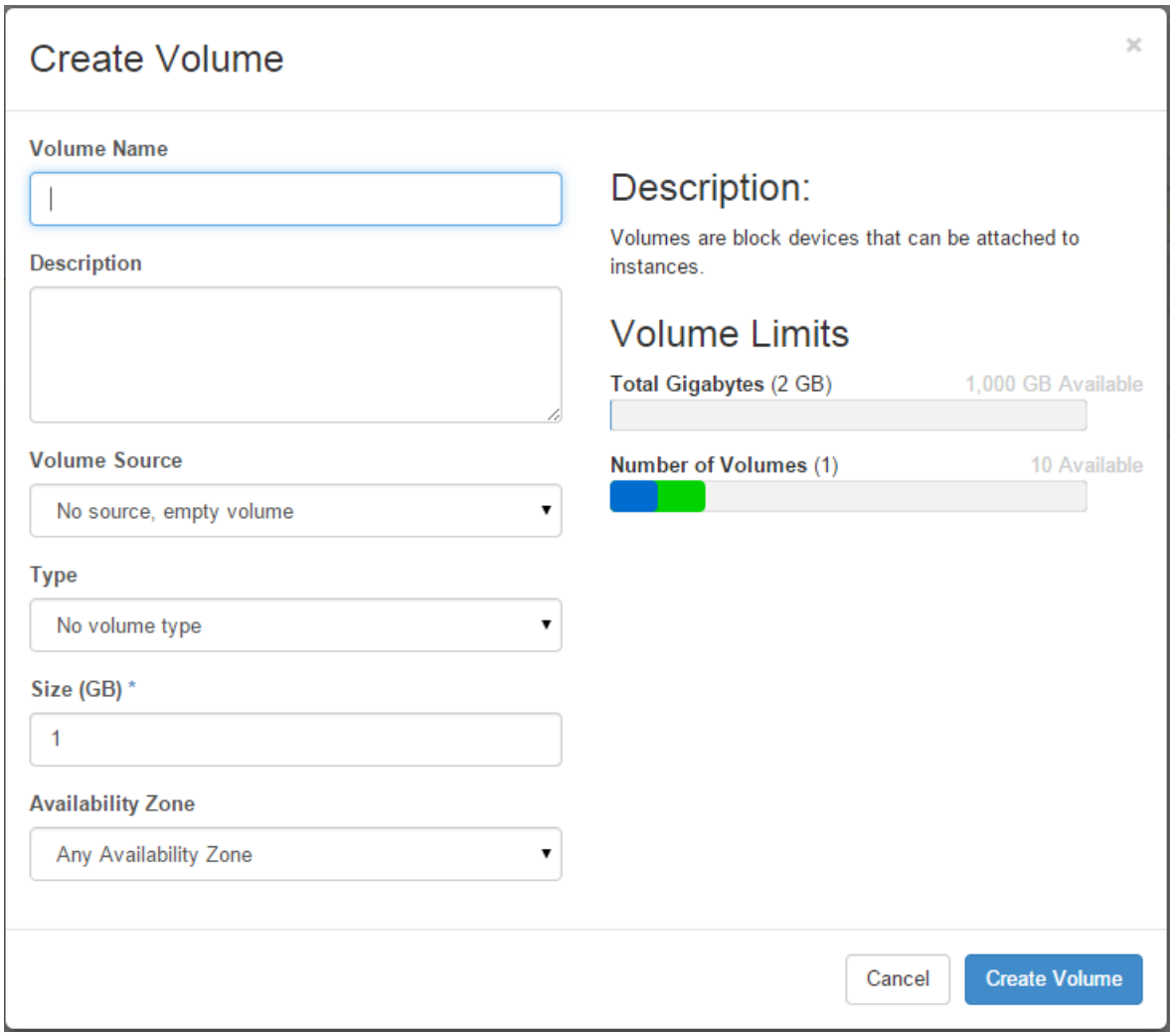
Στην καρτέλα details ο χρήστης ζητείται να δώσει στο νέο instance ένα όνομα το οποίο θα τον βοηθάει να το ξεχωρίζει από τα υπόλοιπα και να είναι αρκετά περιεκτικό ώστε να αντικατοπτρίζει το περιεχόμενο του (αυτό για πιο εύκολη οργάνωση της δουλειάς του). Από την λίστα Flavor καλείται να επιλέξει ανάμεσα σε προεπιλογές οι οποίες αυστηρά καθορίζουν τις απαιτήσεις που θα έχει η δημιουργία αυτού του instance σε πόρους. Έτσι αναλόγως την επιλογή (m1.nano/micro/tiny/small/medium/large/xlarge) θα δοθεί αυτόματα από το σύστημα ένας προεπιλεγμένος κατάλληλος αριθμός εικονικών επεξεργαστών, μνήμης προσπέλασης (RAM) και αποθηκευτικού χώρου. Ο χρήστης μπορεί

επίσης να επιλέξει των αριθμό παρόμοιων τέτοιων instances που θέλει να εκτελέσει καθώς και την πηγή τους η οποία μπορεί να είναι είτε από εικόνα που έχουμε ήδη ανεβασμένη στον διακομιστή νέφους, από στιγμιότυπο (snapshot) κάποιου προηγούμενου instance, ή από κάποιον αποθηκευτικό χώρο που έχουμε δημιουργήσει και φιλοξενεί ένα ήδη έτοιμο instance το οποίο είναι έτοιμο προς εκτέλεση.

Προαιρετικά και ανάλογα με τις ρυθμίσεις που έχει κάνει ο διαχειριστής στο νέφος ο χρήστης μπορεί να χρειαστεί να κάνει επιλογές και στις καρτέλες «Access & Security» ή/και «Networking» για να ρυθμίσει την δικτυακή επικοινωνία του νέου instance που δημιουργεί καθώς και το ποιοι κανόνες θα διέπουν την κίνηση από και προς αυτό. Στην δική μας υλοποίηση συνίσταται αυτά να αφήνονται ως έχουν στις προκαθορισμένες (default) ρυθμίσεις.

6.3.3.2 Δημιουργία ενός νέου αποθηκευτικού χώρου

Αντίστοιχα με τα instances ο χρήστης μπορεί να δημιουργήσει προσωπικά κομμάτια αποθηκευτικού χώρου (volumes) τα οποία θα δρουν ως σκληρός δίσκος για το εκάστοτε instance εάν και εφόσον το επιθυμεί. Αυτά τα κομμάτια μπορούν να προσκολληθούν (attach) ή να αποκολληθούν (detach) από κάποιο instance κατά το δοκούν. Για την δημιουργία ενός volume ο χρήστης πρέπει να μεταβεί στην καρτέλα «Volumes» του μενού «Compute». Εκεί μπορεί να εκκινήσει την διαδικασία δημιουργίας πατώντας στο κουμπί «Create Volume».



6.11 Επιλογές κατά την δημιουργία Volume

Όμοια με προηγουμένως ο χρήστης πρέπει να δώσει στον αποθηκευτικό του χώρο μια ευδιάκριτη ονομασία και μια προαιρετική περιγραφή. Μια ακόμα βασική επιλογή είναι σαφώς το μέγεθος του αποθηκευτικού χώρου το οποίο καλείται να ορίσει από το πεδίο «Size (GB)» ενώ αναλόγως τις ανάγκες της δουλειάς του μπορεί να χρειαστεί να αλλάξει την πηγή δημιουργίας του αποθηκευτικού χώρου ο οποίος μπορεί να είναι ένα νέο κενό volume, να προέρχεται από ένα παλαιότερο (ήδη υπάρχον) ή να προέρχεται από κάποια εικόνα λειτουργικού συστήματος. Για την δική μας υλοποίηση η επιλογή ενός νέου κενού αποθηκευτικού χώρου (No source, empty volume) είναι η προτιμότερη.

Επίλογος

Στο κεφάλαιο αυτό είδαμε το περιβάλλον υλοποίησης του LaaS. Όπως γίνεται αντιληπτό, μέσω του Openstack, μας δίνεται η δυνατότητα διαχείρισης των εικόνων και των αποθηκευτικών χώρων με τα εργαλεία που μας διαθέτει. Είδαμε επίσης, τον εξοπλισμό που χρησιμοποιήσαμε καθώς και τους clients για την πρόσβαση στο σύστημα. Είναι πολύ μεγάλο πλεονέκτημα ότι η πρόσβαση στο σύστημα μπορεί να πραγματοποιηθεί μέσω ενός απλού φυλλομετρητή ενός υπολογιστή, ενός smartphone ή tablet. Στο ακόλουθο και τελευταίο κεφάλαιο που ακολουθεί θα παρατεθούν τα συμπεράσματα καθώς και οι μελλοντικές επεκτάσεις που θα μπορούσε να έχει το σύστημα που αναπτύξαμε σε σχέση με τα ήδη υπάρχοντα συστήματα ή συστήματα που θα μπορούσαν να δημιουργηθούν για να το πλαισιώσουν.

7 Συμπεράσματα και μελλοντικές επεκτάσεις

Εισαγωγή

Στο παρόν και τελευταίο κεφάλαιο αυτής της πτυχιακής εργασίας θα παραθέσουμε τα συμπεράσματα μας γύρω από την σχεδίαση, την ανάπτυξη και την υλοποίηση του Lab as a Service. Τα συμπεράσματα αυτά θα είναι συνέπεια της τριβής και της εμπειρίας με το αντικείμενο που αναλύσαμε στα προηγούμενα κεφάλαια. Στην συνέχεια του κεφαλαίου θα παρατεθούν κάποιες σκέψεις και προτάσεις για μελλοντικές επεκτάσεις και προσθήκες που θα μπορούσαν να υλοποιηθούν και να εφαρμοστούν. Επίσης, θα γίνουν και προτάσεις για μελλοντικές υλοποιήσεις υπηρεσιών, με την χρήση υπολογιστικού νέφους, εντός του ιδρύματος που θα μπορούσαν να αλληλεπιδρούν με το LaaS.

7.1 Συμπεράσματα

Τα συμπεράσματα μας χωρίζονται κατά κάποιο τρόπο σε τρεις ξεχωριστές συνιστώσες. Τα διαχωρίσαμε ως προς τις δυνατότητες του τωρινού συστήματος, ως προς την λειτουργικότητα του και τέλος, στην αξία που προσθέτει στις τωρινές υπηρεσίες του ιδρύματος.

7.1.1 Συμπεράσματα ως προς τις δυνατότητες του συστήματος

Τα συμπεράσματα στα οποία καταλήξαμε έγιναν με γνώμονα τον εξοπλισμό τον οποίο χρησιμοποιήσαμε. Ο εξοπλισμός και η υπολογιστική ισχύ παίζουν πολύ σημαντικό ρόλο σε οποιοδήποτε σύστημα. Το Las as a Service δεν θα μπορούσε, προφανώς, να αποτελεί εξαίρεση. Έτσι οι δυνατότητες με την χρήση ενός και μόνο μηχανήματος ήταν αν, όχι περιορισμένες ,τότε σίγουρα συγκεκριμένες.

Αυτό συμβαίνει διότι το Openstack έχει κάποια όρια σε εικονικά χαρακτηριστικά όπως είδαμε και στις εικόνες 5.13 και 5.18 με την προεπισκόπηση και τις προεπιλογές, αντίστοιχα.

Χαρακτηριστικά, έχουμε την δυνατότητα διαχείρισης:

- Δέκα διαφορετικών **Instances**
- Είκοσι **VCPUs**

- 51200 MB(51GB) **RAM**

Τα νούμερα αυτά μπορεί να φαίνονται μεγάλα, αλλά εξαρτώνται από τις διαστάσεις και τις πτυχές που θέλουμε να δώσουμε στο σύστημα μας. Το συμπέρασμα μας πάνω σε αυτό το κομμάτι είναι ότι αν το LaaS πρέπει να είναι λειτουργικό δεν μπορεί να βασιστεί σε μια υλοποίηση με την χρήση ενός μόνο μηχανήματος.

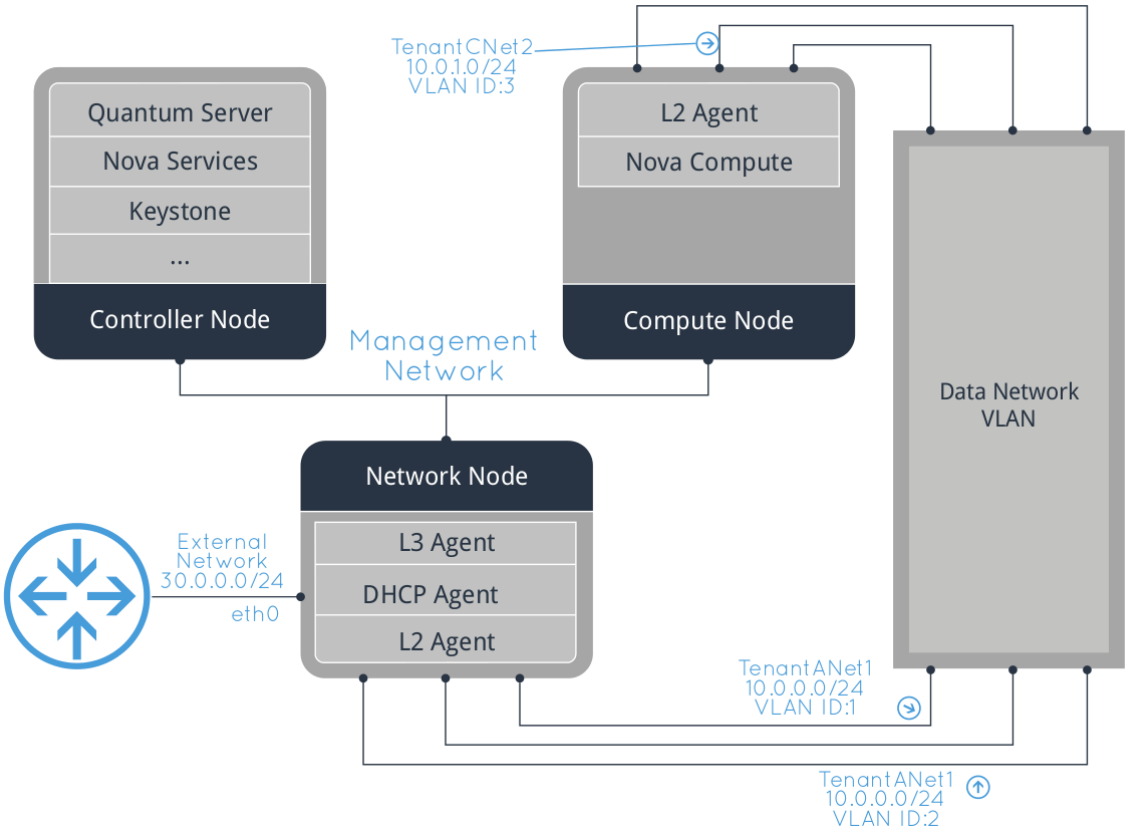
Απόρροια του παραπάνω είναι τα συμπεράσματα που εξάγαμε ως προς την λειτουργικότητα που θα παρατεθούν αμέσως παρακάτω.

7.1.2 Συμπεράσματα ως την λειτουργικότητα του συστήματος

Στην υποενότητα αυτή θα παραθέσουμε τα συμπεράσματα μας ως προς την λειτουργικότητα του συστήματος. Πιο αναλυτικά, τι θα μπορούσε να γίνει έτσι ώστε το εγχείρημα αυτό να γίνει λειτουργικό.

Το βασικότερο συμπέρασμα έχει να κάνει με την υλοποίηση, και ειδικότερα με την χρήση παραπάνω υλικοτεχνικού εξοπλισμού. Υπάρχουν πολλές διαφορετικές υλοποιήσεις για την υλοποίηση του Openstack και κατ' επέκταση του LaaS. Άλλες είναι πιο εξειδικευμένες και πολύπλοκες και ενώ άλλες πιο απλές. Γενικά προτείνεται η χρησιμοποίηση τριών διαφορετικών μηχανών για την υλοποίηση ενός λειτουργικού συστήματος με το Openstack. Αρχικά, η βάση του κάθε μηχανήματος θα είναι προφανώς το ίδιο λειτουργικό σύστημα, λόγου χάρη Linux Ubuntu Server. Στην συνέχεια κάθε μηχάνημα (κόμβος) θα έχει εγκατεστημένο το Openstack απλά θα έχει διαφορετικές ρυθμίσεις από τα άλλα.

Στην παρακάτω εικόνα φαίνεται ένα παράδειγμα υλοποίησης με την χρησιμοποίηση τριών κόμβων.



7.1 Υλοποίηση Openstack με τρεις κόμβους

Σύμφωνα με το παραπάνω θα υπάρχουν 3 διαφορετικοί κόμβοι. Ο πρώτος κόμβος, Controller Node, θα έχει τον ρόλο του ελεγκτή, φιλοξενώντας τις υπηρεσίες Nova, Keystone, Neutron κτλ. Ο δεύτερος κόμβος, Compute Node, θα έχει τον ρόλο του επεξεργαστικού κόμβου φιλοξενώντας το Nova Compute και τον L2 Agent για να εξυπηρετεί την επικοινωνία 2ου επιπέδου. Ο τρίτος και τελευταίος κόμβος, Network Node, θα εξυπηρετεί την δικτύωση σε 3ο επίπεδο φιλοξενώντας τους L3, L2 και DHCP Agents.

7.1.3 Προστιθέμενη αξία

Η υπηρεσία αυτή προσθέτει αξία στις ήδη υπάρχουσες υπηρεσίες που παρέχονται από το ίδρυμα, όπως είναι παροχή αποθηκευτικού χώρου στον server **titanas.it.teithe.gr** και η παροχή **προσωπικής βάσης MySQL** στον aetos.

Η αξία της εφαρμογής αυξάνεται και λόγω της καινοτομίας της. Σκοπός της εφαρμογής είναι ο διαμοιρασμός εικόνων λειτουργικών συστημάτων προς τους φοιτητές του τμήματος, οι οποίοι μπορούν να έχουν πρόσβαση σε αυτές τις εικόνες είτε τοπικά είτε απομακρυσμένα με την χρήση ενός κοινού browser.

Τέλος, λόγω του χαρακτηριστικού ότι η εφαρμογή στηρίχθηκε σε λειτουργικό σύστημα και λογισμικό ανοιχτού κώδικα, την κάνει εύκολα επεκτάσιμη σε περίπτωση που κάποιον το επιθυμεί. Οι επεκτάσεις και οι πιθανές αλληλεπιδράσεις με άλλα συστήματα παρατίθενται στην υποενότητα που ακολουθεί.

7.2 Μελλοντικές Επεκτάσεις

Οι μελλοντικές επεκτάσεις για την υπηρεσία μπορούν να διαχωριστούν σε δύο κατηγορίες. Η πρώτη κατηγορία έχει να κάνει με την υπηρεσία και το περιβάλλον αυτό καθεαυτό και η δεύτερη σε συνεργασία με άλλες υπηρεσίες του ιδρύματος που υπάρχουν ή δύναται να αναπτυχθούν.

7.2.1 Επεκτάσεις ως προς την υπηρεσία

Μια από την πιθανές επεκτάσεις και ίσως η πιο σημαντική είναι η αλλαγή του εξοπλισμού που θα χρησιμοποιηθεί για να φιλοξενήσει το περιβάλλον LaaS. Θα μπορούσε να υπάρχει αναβάθμιση στην επεξεργαστική ισχύ, στον αποθηκευτικό χώρο και στην μνήμη του μηχανήματος.

Επίσης, θα μπορούσαμε να επεκτείνουμε και να αυξήσουμε τις εικόνες που θα φιλοξενούνται στο LaaS. Για παράδειγμα, σε περίπτωση που άλλαζε το πρόγραμμα σπουδών και εντασσόταν ένα μάθημα εκμάθησης και προγραμματισμού Android εφαρμογών, το LaaS θα μπορούσε να φιλοξενεί και να διαμοιράζει εικόνες Android λογισμικού όπου οι φοιτητές θα μπορούσαν να ελέγχουν πως συμπεριφέρεται η εφαρμογή τους σε ένα Android περιβάλλον. Στην παρούσα φάση, βέβαια, αυτό δεν είναι ακόμα πραγματοποιήσιμο και λειτουργικό,

αλλά μιας και το Openstack εξελίσσεται διαρκώς στο μέλλον είναι πολύ πιθανό να είναι υλοποιήσιμο.

Μια τεράστια επέκταση θα μπορούσε να είναι η μεταφορά όλων των servers του ιδρύματος πάνω σε εικονικά μηχανήματα που θα φιλοξενούνταν στο LaaS. Αυτό βέβαια, είναι εκτός του φάσματος τις εξομοίωσης ενός εργαστηριακού περιβάλλοντος, μιας και εντάσσεται στην γενική έννοια του Virtualization. Πιο συγκεκριμένα θα μπορούσε να δημιουργηθεί ένα Cloud με πολύ δυνατά φυσικά χαρακτηριστικά, το οποίο θα φιλοξενεί πολλούς εικονικούς servers που θα αντιστοιχούν στους τωρινούς πχ. aetos, hydra, titanas, erodios κτλ. Ακόμα εκτός από αυτούς θα μπορεί να φιλοξενεί όπως είναι λογικό και έναν server LaaS που θα εξομοιώνει τα εργαστηριακά περιβάλλοντα όπως έχουμε ήδη περιγράψει.

Περαιτέρω επέκταση της εφαρμογής θα μπορούσε να είναι η αλλαγή του γραφικού περιβάλλοντος του Horizon(Dashboard) σε ένα περιβάλλον που θα δημιουργούνταν από το ίδρυμα για δική του χρήση. Με αυτήν την αλλαγή θα μπορούσε το γραφικό περιβάλλον να προβάλλει διαφορετικές πληροφορίες από αυτές που δίνει το Openstack και να παρέχει και διαφορετικές υπηρεσίες και ενέργειες μια και τα SDKs[50] (Software Development Kits) του είναι ανοιχτά προς χρήση και επεξεργασία.

7.2.2 Επεκτάσεις ως προς την αλληλεπίδραση με άλλες υπηρεσίες του τμήματος

Η υλοποίηση του LaaS μπορεί να αλληλεπιδρά με υπηρεσίες του ιδρύματος που μπορούν να σχεδιαστούν βασισμένες στο στην τεχνολογία του υπολογιστικού νέφους. Αν όλες οι υπηρεσίες αυτές δημιουργηθούν στο νέφος τότε θα επωφελούνται από τα πλεονεκτήματα που το χαρακτηρίζουν.

Μια τέτοια πλατφόρμα που θα μπορούσε να αναπτυχθεί θα ήταν ένα **Storage as a Service**[51] περιβάλλον. Η υπηρεσία αυτή θα μπορούσε να προσφέρει αποθηκευτικό χώρο στους φοιτητές του τμήματος πληροφορικής μέσω της τεχνολογίας του υπολογιστικού νέφους. Υπάρχουν πολλές τέτοιες πλατφόρμες που ήδη παρέχουν αυτή την υπηρεσία π.χ. το Dropbox[52], το Box[53] κτλ . Με μια τέτοια υπηρεσία ο φοιτητής θα μπορεί να αποθηκεύει τα προσωπικά του δεδομένα, με ασφάλεια, στο υπολογιστικό νέφος της σχολής και να έχει πρόσβαση από οπουδήποτε επιθυμεί.

Άλλη μια εφαρμογή που θα έβρισκε αντίκρισμα και θα ήταν πολύ αποτελεσματική θα ήταν η λειτουργία Βάσεων Δεδομένων στο νέφος, διαδεδομένο ως **Database as a Service (DBaaS)**[54]. Το DBaaS θα μπορεί να λειτουργεί διανέμοντας τις βάσεις δεδομένων που υπάρχουν στο εσωτερικό του με τις υπόλοιπες πλατφόρμες που τις αιτούνται και επίσης να παρέχει προσωπικές βάσεις δεδομένων προς χρήση στους φοιτητές όπως ακριβώς συμβαίνει με την παρούσα υλικοτεχνική δομή του ιδρύματος.

Ακόμα μια υπάρχουσα εφαρμογή που θα μπορούσε να μεταφερθεί στην υποδομή του υπολογιστικού νέφους είναι η υπηρεσία της αυθεντικοποίησης και ειδικότερα το LDAP. Χαρακτηριστικά όπως αναφέρεται και στους κύκλους του IT,ως **LDAP-as-a-Service**[55]. Αυτή η εφαρμογή θα μπορεί να διαμοιράζει τις υπηρεσίες αυθεντικοποίησης σε όλες τις υπόλοιπες που απαιτούν αυθεντικοποίηση από τους χρήστες τους, φοιτητές, καθηγητές κοκ.

Γενικά, η μεταφορά των υπηρεσιών στο νέφος μπορεί να απλουστεύσει και να βελτιώσει πολύ τα πράγματα στον τομέα της παροχής των υπηρεσιών προς τους άμεσα ενδιαφερόμενους (φοιτητές, καθηγητές, διαχειριστές κτλ) και να αναπτύξει και να εξελίξει την υλικοτεχνική υπόσταση του τμήματος μας. Επίσης, αν και εφόσον είναι δυνατόν να συμβάλει στην μεταφορά της υποδομής όλου του Αλεξάνδρειου ΤΕΙ, από την τωρινή που βρίσκεται , στο νέφος, με σκοπό όχι μόνο την ευκολότερη διαχείριση αλλά και την ανάπτυξη και εξέλιξη όλου του ιδρύματος σαν οργανισμό.

8 Ενδεικτική Βιβλιογραφία και Αναφορές

Ενδεικτική Βιβλιογραφία

- Αλεξόπουλος, Αριστείδης, Λαγογιάννης, Γεώργιος (2011), *Τηλεπικοινωνίες και δίκτυα υπολογιστών*, Παπάγου, Ελλάδα, Εκδόσεις Γιαλός.
- Γκρίζαλης, Σ., Κάτσικας, Σ., Γκρίζαλης, Δ. (2003), *Ασφάλεια Δικτύων Υπολογιστών*, Αθήνα, Ελλάδα, Εκδόσεις Παπασωτηρίου.
- Arnold, Joe (2014), *OpenStack Swift*, USA, Εκδόσεις O'Reilly Media, Inc.
- Fifield, Tom, Fleming, Diane, Gentle, Anne (2014), *OpenStack Operations Guide*, USA, Εκδόσεις O'Reilly Media, Inc.
- Golden, Bernard (2007), *Virtualization For Dummies*, Εκδόσεις John Wiley and Sons Ltd.
- Hurwitz, J., Bloor, R., Kaufman, M., Halper, F. (2010), *Cloud Computing For Dummies*, Indiana, USA, Εκδόσεις Wiley Publishing, Inc.
- Jackson, K., Bunch, C. (2013), *Openstack Cloud Computing Cookbook Second Edition*, Birmingham, UK, Εκδόσεις Packt Publishing Ltd.
- Mather, T., Kumaraswamy, S., Latif, S. (2009), *Cloud Security and Privacy*, USA, Εκδόσεις O'Reilly Media, Inc.
- Pepple, Ken (2014), *Deploying OpenStack*, USA, Εκδόσεις O'Reilly Media, Inc.
- Wolf, C., Halter, E. (2008), *Virtualization: From the Desktop to the Enterprise*, Εκδόσεις Apress.

Αναφορές

1. Υπολογιστικό Νέφος – Wikipedia – http://en.wikipedia.org/wiki/Cloud_computing
2. Virtual private network(VPN) – Wikipedia – http://en.wikipedia.org/wiki/Virtual_private_network
3. Openstack– Wikipedia – http://en.wikipedia.org/wiki/OpenStack
4. Linux– Wikipedia – http://en.wikipedia.org/wiki/Linux
5. Ubuntu – Wikipedia - http://en.wikipedia.org/wiki/Ubuntu_(operating_system)
6. Λειτουργικό Σύστημα –Wikipedia - http://el.wikipedia.org/wiki/Λειτουργικό_σύστημα
7. Virtualization- Wikipedia- http://el.wikipedia.org/wiki/Εικονικοποίηση
8. Infrastructure as a Service – Wikipedia- http://en.wikipedia.org/wiki/Cloud_computing#Infrastructure_as_a_service_.28IaaS.29
9. Platform as a Service- Wikipedia- http://en.wikipedia.org/wiki/Platform_as_a_service
10. Software as a Service- Wikipedia- http://en.wikipedia.org/wiki/Software_as_a_service
11. Private Cloud- Wikipedia- http://en.wikipedia.org/wiki/Cloud_computing#Private_cloud
12. Public Cloud – Wikipedia - http://en.wikipedia.org/wiki/Cloud_computing#Public_cloud
13. Hybrid Cloud– Wikipedia – http://en.wikipedia.org/wiki/Cloud_computing#Hybrid_cloud
14. Παραπληροφόρηση και ανησυχίες γύρω από το υπολογιστικό νέφος - http://www.rackspace.co.uk/cloud-computing/myths
15. Διασφάλιση Λογισμικού http://www.dmmserver.com/DialABook/978/047/046/9780470461907.html

16. CloudStack – Wikipedia -
http://en.wikipedia.org/wiki/Apache_CloudStack
17. Eucalyptus – Wikipedia -
http://en.wikipedia.org/wiki/Eucalyptus_(software)
18. vCloud- Wikipedia -
http://en.wikipedia.org/wiki/VCloud_Air
19. API – Wikipedia-
http://en.wikipedia.org/wiki/Application_programming_interface
20. Single Sign On (SSO) - Wikipedia -
http://en.wikipedia.org/wiki/Single_sign-on
21. Single Log Out (SLO) -
https://tuakiri.ac.nz/confluence/pages/viewpage.action?pageId=9601502
22. SAML -
https://cwiki.apache.org/confluence/display/CLOUDSTACK/SAML+2.0+Plugin
http://en.wikipedia.org/wiki/Security_Assertion_Markup_Language
23. XML – Wikipedia –
http://en.wikipedia.org/wiki/XML
24. Lightweight Directory Access Protocol (LDAP) – Wikipedia -
http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol
25. Active Directory (AD) - Wikipedia -
http://en.wikipedia.org/wiki/Active_Directory
26. Xen - Wikipedia -
http://en.wikipedia.org/wiki/Xen
27. Kernel-based Virtual Machine (KVM) – Wikipedia -
http://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine
28. ESXi – Wikipedia -
http://en.wikipedia.org/wiki/VMware_ESX
29. Amazon Web Service – Wikipedia -
http://en.wikipedia.org/wiki/Amazon_Web_Services
30. Virtual Computing Lab – VCL -
http://www.ibm.com/developerworks/library/ws-vcl/
31. The Openstack Foundation
http://www.openstack.org/
32. MySQL - Wikipedia-
http://en.wikipedia.org/wiki/MySQL
33. Message Queuing Service – Wikipedia -
http://en.wikipedia.org/wiki/Message_queuing_service

34. Hypervisor – Wikipedia - http://en.wikipedia.org/wiki/Hypervisor
35. Keystone – https://wiki.openstack.org/wiki/Keystone http://en.wikipedia.org/wiki/OpenStack#Identity_Service_.28Keystone.29
36. Glance – https://wiki.openstack.org/wiki/Glance http://en.wikipedia.org/wiki/OpenStack#Image_Service_.28Glance.29
37. Nova – https://wiki.openstack.org/wiki/Nova http://en.wikipedia.org/wiki/OpenStack#Compute_.28Nova.29
38. Neutron Networking – https://wiki.openstack.org/wiki/Neutron http://en.wikipedia.org/wiki/OpenStack#Networking_.28Neutron.29
39. Horizon (Dashboard) – https://wiki.openstack.org/wiki/Horizon http://en.wikipedia.org/wiki/OpenStack#Dashboard_.28Horizon.29
40. Cinder – https://wiki.openstack.org/wiki/Cinder http://en.wikipedia.org/wiki/OpenStack#Block_Storage_.28Cinder.29
41. Υλοποίηση Keystone – Openstack Docs - http://docs.openstack.org/juno/install-guide/install/apt/content/keystone-install.html
42. Image – Wikipedia - http://en.wikipedia.org/wiki/System_image
43. Υλοποίηση Glance – Openstack Docs - http://docs.openstack.org/juno/install-guide/install/apt/content/glance-install.html
44. Υλοποίηση Ubuntu Εικόνας - Openstack Docs - http://docs.openstack.org/image-guide/content/ubuntu-image.html
45. Υλοποίηση Windows Εικόνας - Openstack Docs - http://docs.openstack.org/image-guide/content/windows-image.html
46. Υλοποίηση Nova – Openstack Docs- http://docs.openstack.org/juno/install-guide/install/apt/content/ch_nova.html
47. Υλοποίηση Neutron – Openstack Docs - http://docs.openstack.org/juno/install-guide/install/apt/content/section_neutron-networking.html
48. Υλοποίηση Cinder – Openstack Docs- http://docs.openstack.org/juno/install-guide/install/apt/content/ch_cinder.html
49. Υλοποίηση Horizon – Openstack Docs - http://docs.openstack.org/juno/install-guide/install/apt/content/install_dashboard.html

50. Openstack SDKs- Openstack Wiki - https://wiki.openstack.org/wiki/SDKs
51. Storage as a Service – Wikipedia- http://en.wikipedia.org/wiki/Storage_as_a_service
52. Dropbox – Wikipedia - http://en.wikipedia.org/wiki/Dropbox_(service)
53. Box – Wikipedia - http://en.wikipedia.org/wiki/Box_%28company%29
54. Database as a Service (DBaaS) - http://www.scaledb.com/dbaas-database-as-a-service.php
55. LDAP as a Service - https://jumpcloud.com/blog/ldap-as-a-service/