

Cloud Computing & OpenStack

Επισκόπηση και Υλοποίηση

Σαββαΐδης Ιωάννης

A.M. 8/07

Παναγιώτου Αλέξανδρος

A.M. 119/07

Θεσσαλονίκη 2012



Τμήμα Εφαρμοσμένης Πληροφορικής

Πτυχιακή εργασία των φοιτητών: Σαββαΐδη Ιωάννη και Παναγιώτου Αλέξανδρου

με τίτλο: Cloud Computing & OpenStack, Επισκόπηση και Υλοποίηση

Επιβλέπων Καθηγητής: Μαργαρίτης Κωνσταντίνος

Ευχαριστίες

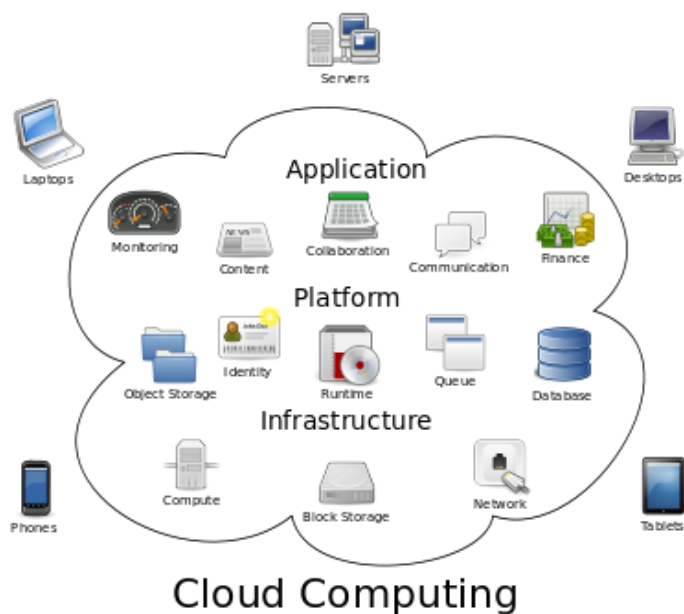
Με την ολοκλήρωση της πτυχιακής εργασίας, θα θέλαμε να ευχαριστήσουμε θερμά τον καθηγητή κύριο Μαργαρίτη Κωνσταντίνο για την ευκαιρία που μας έδωσε να αναλύσουμε το παρόν θέμα αλλά και για την καθοδήγηση του. Επίσης θέλουμε να ευχαριστήσουμε τον κύριο Γεώργιο Φώτη , για την πολύτιμη βοήθεια του σε όλα τα τεχνικά προβλήματα.

Περιεχόμενα

Ευχαριστίες.....	2
1. Εισαγωγή	4
2. Cloud Computing	10
1. Τα μεγάλα ονόματα στο Cloud Computing	10
2. Υλικό και υποδομή.....	15
3. Αποθήκευση Στο «Νέφος»	20
4. Το Cloud Computing στις επιχειρήσεις	23
5. Το Μέλλον του Cloud Computing	27
3. Εισαγωγή στο OpenStack	28
1. Υπηρεσία Υπολογιστικής Ισχύος – Nova.....	28
2. Υπηρεσία Εικόνων - Glance	31
3. Υποδομή Αποθήκευσης – Swift.....	31
4. Υπηρεσία ταυτοποίησης – Keystone	33
5. Γραφική διεπαφή διαχειριστή – Horizon	36
6. Αρχιτεκτονική του OpenStack.....	37
4. Εγκατάσταση του συστήματος	39
1. Server1	40
2. Server2	63
3. Client1.....	66
5. Διαχείριση Εικόνων	69
6. Διαχείριση των Instances	71
7. Διεπαφή Χρήστη (Dashboard)	77
Βιβλιογραφία	82

1. Εισαγωγή

Το cloud computing (Υπολογιστική Νέφος) βρίσκεται παντού. Σε όλα τα μέσα δικτύωσης, τεχνολογικά περιοδικά και ιστολόγια. Δεν υπάρχει όμως μια συγκεκριμένη άποψη, αλλά πολλές οι οποίες μας δυσκολεύουν να βρούμε έναν ακριβή όρο. Γενικά ο όρος cloud computing συνδέεται με το internet, αλλά στο παρασκήνιο συμβαίνουν πολλά περισσότερα. Το cloud computing είναι μια κατασκευή που επιτρέπει την απομακρυσμένη πρόσβαση σε υπηρεσίες εφαρμογών, λογισμικού και αποθηκευτικού χώρου χωρίς ο τελικός χρήστης να γνωρίζει σε ποια γεωγραφική θέση βρίσκονται αυτές.



Εικόνα 1. Τα συστατικά του Cloud Computing¹

Επειδή τίποτα δεν είναι τέλει έτσι και στο cloud computing υπάρχουν προβλήματα, όπως η διακοπή της σύνδεσης στο internet ή την αδυναμία ολοκλήρωσης κάποιων εφαρμογών οι οποίες είναι γεωγραφικά διασκορπισμένες.

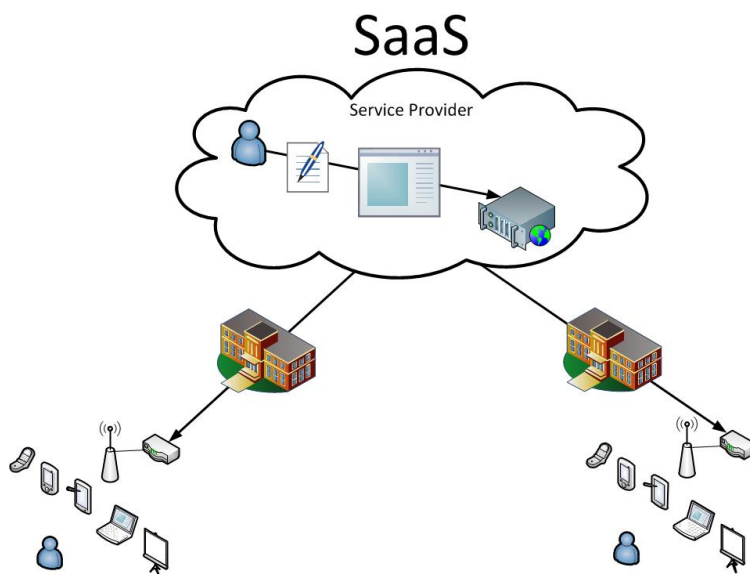
¹ Εικόνα από http://en.wikipedia.org/wiki/Cloud_computing

Μια λύση cloud computing αποτελείται από : πελάτες, το κέντρο δεδομένων και κατανεμημένους διακομιστές. Κάθε ένα από αυτά τα στοιχεία παίζει καθοριστικό ρόλο στην εφαρμογή ενός «νέφους».

- **Πελάτες** : Είναι οι συσκευές με τις οποίες ο τελικός χρήστης αλληλεπιδρά με το «νέφος».
- **Κέντρο δεδομένων** : Είναι το σύνολο των διακομιστών στους οποίους φιλοξενοούνται οι εφαρμογές. Μπορεί να είναι φυσικοί ή εικονικοί.
- **Κατανεμημένοι διακομιστές** : Συνήθως βρίσκονται σε διαφορετικές θέσεις αλλά ενεργούν σαν να βρίσκονται ο ένας δίπλα στον άλλο.

Οι υπηρεσίες που μπορεί να προσφέρει μια λύση cloud computing είναι:

- **Software as a Service (SaaS) Λογισμικό ως Υπηρεσία**

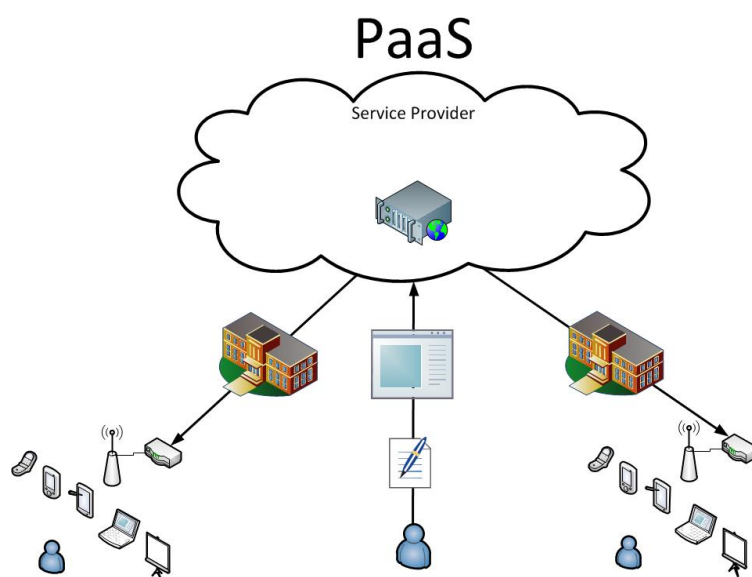


Εικόνα 2. Λογισμικό ως Υπηρεσία (SaaS)²

² Εικόνα από <http://edutechassociates.net/2011/02/23/cloud-watching-1-cloud-101/>

Το λογισμικό παρέχεται ως υπηρεσία στον χρήστη, ο οποίος μπορεί να έχει πρόσβαση σ'αυτό από οποιαδήποτε συσκευή μέσω σύνδεσης στο Web. Το πλεονέκτημα είναι ότι ο χρήστης δεν αγοράζει την εφαρμογή, και πληρώνει μόνο για την προσωρινή χρήση. Ορισμένες από αυτές τις εφαρμογές είναι εντελώς δωρεάν όπως το Hotmail, το Google Apps, Skype, και πολλές άλλες εφαρμογές του Web 2.0.

- Platform as a Service (PaaS) Πλατφόρμα ως Υπηρεσία

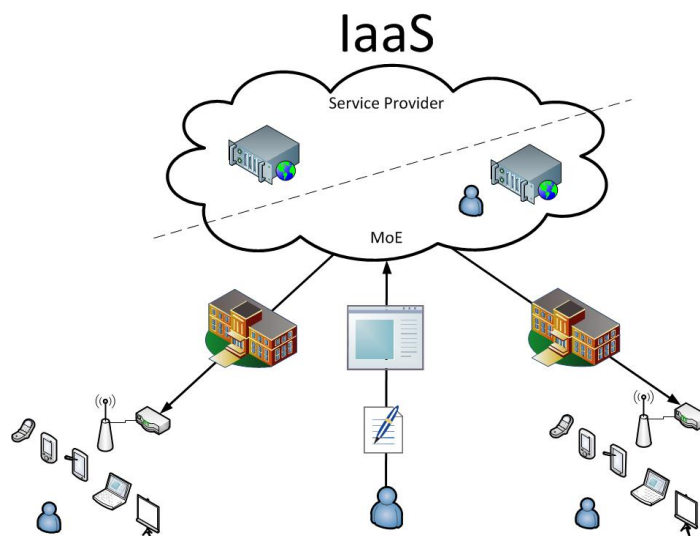


Εικόνα 3. Πλατφόρμα ως Υπηρεσία (PaaS)³

Σε αντίθεση με πρώτη κατηγορία, που απευθύνεται στους τελικούς χρήστες, αυτή εξυπηρετεί κυρίως τους προγραμματιστές. Μια πλατφόρμα μπορεί να παρέχει ένα σύνολο των προγραμμάτων, εργαλείων, βασικών υπηρεσιών και βιβλιοθηκών με σκοπό τη δημιουργία, δοκιμή και ανάπτυξη εφαρμογών. Τα πιο γνωστά παραδείγματα – Google App Engine και Windows Azure.

³ Εικόνα από <http://edutechassociates.net/2011/02/23/cloud-watching-1-cloud-101/>

- Infrastructure as a Service (IaaS) Υποδομή ως Υπηρεσία



Εικόνα 4. Υποδομή ως Υπηρεσία (IaaS)⁴

Μια εικονική διάθεση υποδομής, όπως διακομιστές, αποθηκευτικός χώρος, υποδομές δικτύου, λειτουργικά συστήματα, τεχνολογίες εξομοίωσης και συστήματα διαχείρισης αρχείων. Είναι δηλαδή μια μορφή φιλοξενίας. Ο πάροχος IaaS συντηρεί το υλικό και διαχειρίζεται τις υπηρεσίες. Web Services της Amazon για παράδειγμα.

Αυτό που κάνει το cloud computing είναι να προσφέρει εφαρμογές σε τελικούς χρήστες σερβίροντας τις στις συσκευές τους. Υπάρχουν πολλές ευρείας χρήσεως εφαρμογές αλλά διαφέρουν στην αποθήκευση και τις βάσεις δεδομένων.

Μια από τις χρήσεις του cloud computing είναι η αποθήκευση. Ο προμηθευτής μισθώνει τον χώρο και δεν χρειάζεται η αγορά εξοπλισμού.

Οι βάσεις δεδομένων είναι αποθήκες πληροφοριών με συνδέσεις μέσα στις πληροφορίες που βοηθούν στην αναζήτηση δεδομένων. Οι κατανεμημένες βάσεις δεδομένων όπως το simple DB της Amazon διαχέουν τις πληροφορίες σε υλικό που βρίσκεται σε διασκορπισμένες θέσεις ενώ φαίνεται σαν να βρίσκονται σε μια.

⁴ Εικόνα από <http://edutechassociates.net/2011/02/23/cloud-watching-1-cloud-101/>

Μέσα στους πρώτους που χρησιμοποίησαν το «νέφος» είναι και οι Amazon, Microsoft και Google.

Η Amazon ήταν από τις πρώτες επιχειρήσεις που πρόσφερε υπηρεσίες cloud computing στο κοινό. Οι υπηρεσίες που προσφέρει είναι:

- Το **Elastic Compute Cloud (EC2)**: Προσφέρει εικονικούς υπολογιστές και πρόσθετη υπολογιστική ισχύ.
- Το **Simple Storage Service (S3)**: Επιτρέπει την αποθήκευση μέχρι 5GB σε μέγεθος στην εικονική υπηρεσία αποθήκευσης της Amazon.
- Το **Simple Queue Service (SQS)**: Επιτρέπει στους υπολογιστές να μιλούν μεταξύ τους μέσω του API περάσματος μηνυμάτων.
- Και το **SimpleDB**: Μια Web υπηρεσία εκτέλεσης ερωτημάτων σε δομημένα δεδομένα σε πραγματικό χρόνο.

Η λύση cloud computing της Microsoft ονομάζεται Microsoft Azure. Τα βασικά συστατικά της Azure Services Platform είναι:

- **Windows Azure**: Παρέχει υπηρεσίες φιλοξενίας και διαχείρισης και κλιμακωτή αποθήκευση χαμηλού επιπέδου, υπολογιστικές λειτουργίες και δικτύωση.
- **Microsoft SQL Services**: Παρέχει υπηρεσίες βάσεων δεδομένων και δημιουργία αναφορών.
- **Microsoft .NET Services**: Παρέχει υλοποιήσεις βασισμένες σε υπηρεσίες των εννοιών του πλαισίου .NET όπως ροές εργασιών.
- **Live Services**: Χρησιμοποιείται για τον διαμοιρασμό, την αποθήκευση και τον συγχρονισμό εγγράφων, φωτογραφιών και αρχείων μεταξύ συσκευών.
- **Microsoft SharePoint Services και Microsoft Dynamics CRM Services**: Χρησιμοποιείται για επιχειρησιακά περιεχόμενα, συνεργασία και ανάπτυξη λύσεων στο «νέφος».

Η Google προσφέρει το Google App Engine. Το Google App Engine είναι μια πλατφόρμα ως υπηρεσία PaaS για την ανάπτυξη και φιλοξενία Web εφαρμογών στα κέντρα δεδομένων της Google. Προσφέρει αυτόματη κλιμάκωση για μια εφαρμογή, δηλαδή αν η ζήτηση της αυξηθεί τότε η Google αυτόματα διαθέτει περισσότερους πόρους γι' αυτήν. Το Google App Engine είναι δωρεάν μέχρι ένα επίπεδο κατανάλωσης πόρων, μετά από αυτό το επίπεδο χρεώνονται τέλη για επιπλέον αποθηκευτικό χώρο,

εύρος ζώνης και ώρες χρησιμοποίησης της εφαρμογής. Προσφέρει περισσότερους υπολογιστικούς πόρους, σε σχέση με τους άλλους παροχείς, ώστε να είναι πιο εύκολο να δημιουργηθεί μια εφαρμογή, αλλά μπορεί να τρέξει ένα περιορισμένο σύνολο εφαρμογών που δημιουργούνται για μια συγκεκριμένη υποδομή.

2. Cloud Computing

1. Τα μεγάλα ονόματα στο Cloud Computing

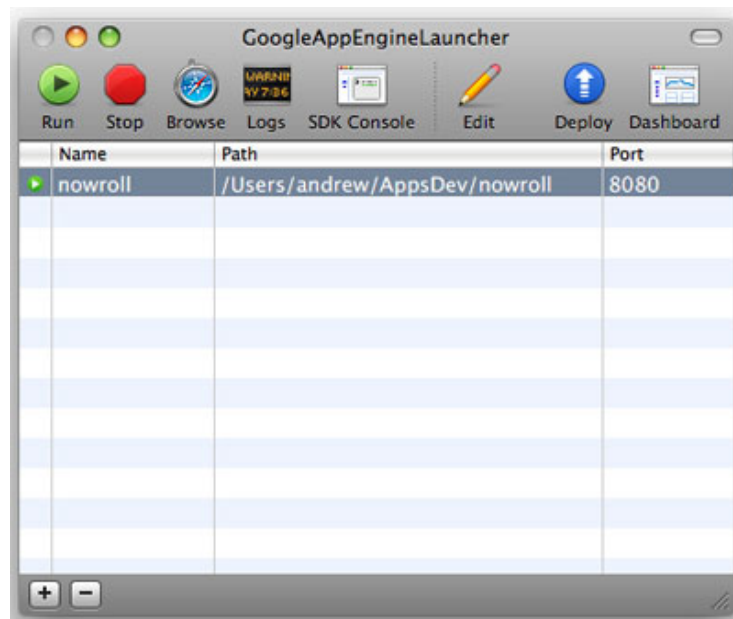
Υπάρχουν πολλοί προμηθευτές του cloud computing αλλά τα μεγαλύτερα ονόματα του cloud computing είναι και τα μεγαλύτερα ονόματα στον κόσμο των υπολογιστών. Μερικά από αυτά τα ονόματα είναι οι Google, Microsoft, Yahoo!, Salesforce, IBM κ.α.

Google

Η Google έχει επενδύσει πολλά στο cloud computing το οποίο είναι μια από τις μεγαλύτερες επιχειρήσεις της. Έχει αναπτύξει μερικά εργαλεία με σκοπό να προσελκύσει πελάτες.

Google App Engine

Το Google App Engine προσφέρει σε προγραμματιστές την δυνατότητα να αναπτύξουν τις δικές τους web εφαρμογές στην ίδια υποδομή με αυτή που ενδυναμώνει τις εφαρμογές της η Google.



Εικόνα 5. Η πλατφόρμα του Google App Engine⁵

Χρησιμοποιώντας το Google App Engine οι προγραμματιστές μπορούν να πετύχουν τα παρακάτω:

- **Σύνταξη κώδικα μια φορά και εγκατάσταση:** Το Google App Engine παρέχει υπολογιστικούς πόρους για web εφαρμογές στους προγραμματιστές έτσι ώστε να μην ξοδεύουν χρόνο και χρήμα στην παρακολούθηση πολλαπλών υπολογιστών.
- **Ευέλικτο στην αυξομείωση της κίνησης:** Όταν μια εφαρμογή γίνει δημοφιλής και αυξηθούν δραματικά η κίνηση σε αυτήν δημιουργείται πρόβλημα και αυτό γίνεται σε όλες τις εφαρμογές, όλων των μεγεθών, όλων των εταιρειών εκτός της Google που κάνει ευκολότερη την κλιμάκωση.
- **Ενοποίηση με άλλες υπηρεσίες της Google:** Είναι αναποτελεσματικό να γράφουν οι προγραμματιστές όλες τις εφαρμογές από το 0. Γιαυτό χρησιμοποιούν τα συστατικά και την βιβλιοθήκη των API της Google που παρέχουν λειτουργικότητα.

⁵ Εικόνα από <http://www.sitepoint.com/rollin-with-google-appengine-80s-style/>

Microsoft

Η Microsoft προσφέρει διάφορες υπηρεσίες για όλες τις επιχειρήσεις, μικρές και μεγάλες. Ένα μέρος των εφαρμογών της είναι παραλλαγές εφαρμογών που ήδη υπάρχουν και χρησιμοποιούνται έτσι δεν είναι δύσκολο να χρησιμοποιηθούν.

Azure Services Platform



Εικόνα 6. Διάγραμμα με τις υπηρεσίες του Microsoft Azure⁶

Η Cloud Computing λύση της Microsoft ονομάζεται Azure Services Platform. Παρέχει λειτουργικότητα για δημιουργία εφαρμογών για άτομα ή μεγάλες επιχειρήσεις. Προσφέρει επίσης λειτουργικό σύστημα «νέφους» και εργαλεία για προγραμματιστές. Οι υπηρεσίες Azure μπορούν να χρησιμοποιηθούν μεμονωμένα ή σε συνδυασμό μεταξύ τους.

Windows Azure

Το Windows Azure είναι ένα Cloud λειτουργικό σύστημα που επιτρέπει την δημιουργία, φιλοξενία και διαχείριση υπηρεσιών. Μπορεί να χρησιμοποιηθεί για προσθήκη Web δυνατοτήτων σε υπάρχουσες εφαρμογές, δημιουργία εφαρμογών και μετακίνηση τους στο Web και γενικά για την διαχείριση Web εφαρμογών αποτελεσματικά και κυρίως ανέξοδα.

⁶ Εικόνα από <http://www.thelab.gr/eidiseis/windows-azure-56483.html>

Windows Live

Το Windows Live είναι ένα σύνολο online υπηρεσιών που κάνουν πιο εύκολο και διασκεδαστικό στους καταναλωτές να επικοινωνούν μεταξύ τους και να μοιράζονται πράγματα με άλλους. Περιλαμβάνει κοινή χρήση φωτογραφιών, ηλεκτρονικό ταχυδρομείο και άμεσα μηνύματα. Το Windows Live Essentials βοηθά τους καταναλωτές να διαχειρίζονται τα περιεχόμενα που έχουν διασκορπισμένα στην συσκευή τους και προσφέρεται δωρεάν για κατέβασμα.

Amazon

Η Amazon είναι αυτή που ίσως ακούγεται περισσότερο από κάθε άλλο στον χώρο του Cloud Computing. Προσφέρει πολλών ειδών υπηρεσίες από αποθήκευση έως πλατφόρμες και βάσεις δεδομένων.

Amazon Elastic Compute Cloud (Amazon EC2)

Το Amazon Elastic Compute Cloud είναι μια υπηρεσία που προσφέρει προσαρμόσιμη χωρητικότητα. Παρέχει ένα Web περιβάλλον που επιτρέπει την εύκολη διαχείριση της χωρητικότητας και τον έλεγχο των υπολογιστικών πόρων. Στον χρήστη φαίνεται σαν ένας απομακρυσμένος υπολογιστής. Το Amazon EC2 δίνει την δυνατότητα να εκτελούνται εφαρμογές βασισμένες σε Windows αλλά στην πλατφόρμα της Amazon.

Amazon SimpleDB

Είναι οι υπηρεσίες βάσεων δεδομένων που προσφέρει η Amazon. Συνεργάζεται με τις άλλες υπηρεσίες της Amazon παρέχει την δυνατότητα αποθήκευσης, επεξεργασίας και χρήσης ερωτημάτων σε σύνολα δεδομένων στο Cloud.

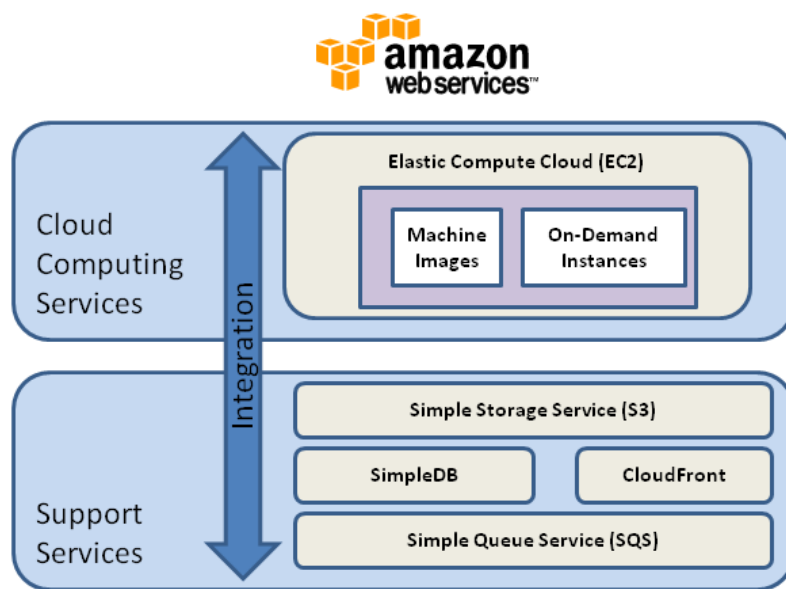
Amazon Simple Storage Service (Amazon S3)

Το Amazon S3 είναι η λύση της Amazon για αποθήκευση στο Internet. Χρησιμοποιεί ένα απλό περιβάλλον Web υπηρεσιών με το οποίο η αποθήκευση και η ανάκτηση πληροφοριών από οπουδήποτε, γίνεται εύκολη.

Elastic Block Store

Το EBS είναι μια σταθερή λειτουργία αποθήκευσης μέσα σε ένα στιγμιότυπο EC2. Παλιά όταν τερματιζόταν το στιγμιότυπο, οτι είχε αποθηκευτεί μέσα σε αυτό χανόταν. Έτσι το EBS δίνει την δυνατότητα στους χρήστες να δημιουργούν νέους

τόμους στους οποίους να τρέχουν τα στιγμιότυπα και στην συνέχεια να τους αποθηκεύουν στο Amazon S3.



Εικόνα 7. Οι υπηρεσίες της Amazon⁷

⁷ Εικόνα από <http://rdn-consulting.com/blog/tag/ec2/>

2. Υλικό και υποδομή

Για να γίνει όσο πιο αποτελεσματική και λειτουργική, μια λύση Cloud Computing πρέπει να υπάρχει το σωστό υλικό και οι σωστές υποδομές. Το υλικό αναφέρεται στις συσκευές που αλληλεπιδρούν με το «νέφος» και οι υποδομή με τις συνδέσεις.

Υπολογιστές-Πελάτες



Εικόνα 8. Συσκευές για σύνδεση στο Cloud Computing⁸

Οι υπολογιστές-πελάτες είναι οι συσκευές των τελικών χρηστών μέσω των οποίων αλληλεπιδρούν με το «νέφος». Υπάρχουν διαφορετικοί τύποι υπολογιστών-πελατών και κάθε ένας προσφέρει διαφορετικό τρόπο αλληλεπίδρασης.

Φορητοί Υπολογιστές-Πελάτες

Υπάρχει μια μεγάλη κλίμακα φορητών Υπολογιστών-Πελατών, από έναν φορητό υπολογιστή έως PDA και έξυπνα τηλέφωνα. Το θετικό με τους φορητούς Υπολογιστές-Πελάτες είναι ότι ο χρήστης μπορεί να έχει πρόσβαση στα δεδομένα του όπου κι αν βρίσκεται, αλλά υπάρχουν προβλήματα ταχύτητας και ασφάλειας σε σχέση με τους σταθερούς Υπολογιστές-Πελάτες όπως μη δυνατή σύνδεση ή απώλεια της συσκευής.

⁸ Εικόνα από <http://www.onbile.com/info/what-cloud-computing-means/>

Λεπτοί Υπολογιστές-Πελάτες

Είναι Υπολογιστές-Πελάτες χωρίς σκληρούς δίσκους, μονάδα DVD-ROM, τρέχουν μόνο εφαρμογές «νέφους» και απλά εμφανίζουν τι υπάρχει στον διακομιστή. Είναι φθηνότεροι από τους παχιούς, πολύ φθηνότεροι στην συντήρηση και καταναλώνουν λιγότερη ενέργεια. Υπάρχει επίσης ένα υψηλό επίπεδο ασφάλειας επειδή δεν αποθηκεύονται δεδομένα σ' αυτούς.

Παχιοί Υπολογιστές-Πελάτες

Είναι οι Υπολογιστές-Πελάτες που ήδη χρησιμοποιούμε. Μπορούν να τρέξουν εφαρμογές «νέφους» αλλά και τοπικά. Πιθανώς κάποιες εφαρμογές πρέπει απλώς να παραμείνουν τοπικά. Σε σχέση με την ασφάλεια είναι πιο τρωτοί από τους λεπτούς αφού τα δεδομένα αποθηκεύονται τοπικά και υπάρχει ενδεχόμενο απώλειας.

Ασφάλεια

Η ασφάλεια είναι το υπ' αριθμόν ένα ζήτημα για το Cloud Computing και αυτό για τον λόγο ότι τα δεδομένα των πελατών αποθηκεύονται σε τρίτους. Η ανησυχία σε σχέση με αυτό είναι λογική, αλλά μην ξεχνάμε και τα πλεονεκτήματα ασφαλείας που υπάρχουν:

- **Διαρροή Δεδομένων:** Για μια εταιρεία με πολλούς υπαλλήλους που χρησιμοποιούν υπολογιστές είναι μεγαλύτερο το όφελος να αποθηκεύονται όλα τα δεδομένα συγκεντρωμένα από το να είναι διασκορπισμένα σε διάφορα μέρη.
- **Καταγραφή:** Η καταγραφή βελτιώνεται. Είναι κάτι που δεν μπορεί να γίνει τοπικά αλλά οι προμηθευτές μπορούν να προσθέσουν όση μνήμη χρειάζεται για να γίνει η καταγραφή.
- **Εργαλεία Παρακολούθησης:** Σε περίπτωση παραβίασης ο παροχέας «νέφους» μπορεί να αποκριθεί στο γεγονός με λιγότερο χρόνο διακοπής απ' ότι τοπικά. Είναι εύκολο να δημιουργηθεί ένας διακομιστής παρακολούθησης online και δεν κοστίζει τίποτα έως ότου χρησιμοποιηθεί.
- **Ανάπτυξη:** Οι προμηθευτές «νέφους» αναπτύσσουν προϊόντα που εφαρμόζονται σε εικονικούς υπολογιστές. Είναι κάτι καινούργιο έτσι οι προμηθευτές ασφαλείας που είναι οραματιστές έχουν πολλές ευκαιρίες ανάπτυξης.
- **Έλεγχος Ασφαλείας:** Είναι πολύ δύσκολο για έναν επαγγελματία μηχανογράφησης να ασφαλίσει το τοπικό δίκτυο. Αυτό αλλάζει γιατί την ασφάλεια την αναλαμβάνει κάποιος άλλος.

Δίκτυο

Για να δώσει το «νέφος» στον καθένα όσα περισσότερα μπορεί, πρέπει να υπάρχουν διαφορετικά επίπεδα σύνδεσης. Κάθε επιχείρηση απαιτεί διαφορετικά πράγματα και έτσι πρέπει να συνδέεται διαφορετικά.

Απλό Δημόσιο Internet

Το απλό δημόσιο Internet είναι το κανάλι που έχουμε όλοι στο γραφείο ή στο σπίτι μας. Είναι η πιο απλή επιλογή για σύνδεση στο «νέφος» ευρυζωνικά ή μέσω τηλεφώνου. Το απλό δημόσιο Internet έχει τα εξής πλεονεκτήματα:

- Υπάρχει μεγάλο ακροατήριο. Όλοι μπορούν να χρησιμοποιήσουν αυτή την λύση.
- Είναι ανεκτικό σε βλάβες.
- Υπάρχουν πολλοί πάροχοι.
- Είναι οικονομικό.

Έχει όμως και μειονεκτήματα:

- Έλλειψη ποιότητας υπηρεσίας
- Πιθανότητα κακής απόκρισης από συνδέσεις μεγάλου χρόνου αναμονής.
- Χρόνος διακοπής που δεν μπορούμε να τον ελέγξουμε.

Το Επιταχυνόμενο Internet

Αυτή η επιλογή μπορεί να ωφελήσει και τον προμηθευτή και τον πελάτη. Η βελτίωση του «νέφος» μπορεί να φτάσει στο 20% με 50% μεταφέροντας από τον διακομιστή λειτουργίες σχετικά με το δίκτυο. Η χρήση SSL και σύνδεσης TCP αφαιρεί επεξεργασία από τους διακομιστές και η δυναμική cache, η συμπίεση και το prefetching δίνουν κατά 50% καλύτερη απόδοση. Αυτή η μέθοδος απευθύνεται στον παροχέα υπηρεσιών αλλά στο τέλος ωφελεί τον πελάτη. Στην μέθοδο αυτή πρέπει να εγκατασταθεί μια συσκευή από την πλευρά του προμηθευτή και ενός προγράμματος από τον πελάτη.

Βελτιστοποιημένη Επικάλυψη Internet

Η μέθοδος αυτή δίνει την δυνατότητα στον τελικό χρήστη να συνδέεται μέσω του δημόσιου Internet αλλά παρέχει βελτίωση στο «νέφος» του παροχέα. Οι βελτιώσεις περιλαμβάνουν:

- Βελτιστοποιημένη δρομολόγηση σε πραγματικό χρόνο.
- Περιεχόμενα που προσπελάζονται συχνά παρέχονται από τοπικές cache.
- Μια σύνοδος SSL μπορεί να σταματήσει ώστε τα πρωτόκολλα να βελτιστοποιηθούν.

Τα μειονεκτήματα της μεθόδου περιλαμβάνουν:

- Είναι έως και τέσσερις φορές πιο ακριβό από το δημόσιο Internet.
- Γίνεται κλείδωμα ενός προμηθευτή εάν η εφαρμογή διανεμηθεί στο δίκτυο του φορέα.

VPN από Θέση σε Θέση

Η τέταρτη επιλογή είναι η σύνδεση μέσω ενός ιδιωτικού δικτύου ευρείας περιοχής(wide area network – WAN). Αυτή διαμόρφωση επιτρέπει εμπιστευτικότητα, εγγυάται εύρος ζώνης και SLA για διαθεσιμότητα, χρόνο αναμονής και απώλεια πακέτων. Στα αρνητικά είναι ότι δεν είναι όσο αξιόπιστο όσο οι άλλες συνδέσεις Internet.

Υπηρεσίες

Υπάρχουν διάφορες υπηρεσίες προς εκτέλεση ανάλογα με τον παροχέα «νέφους» και με αυτά που χρειαζόμαστε. Αυτές οι υπηρεσίες επηρεάζουν το πως θα στηθεί η υποδομή.

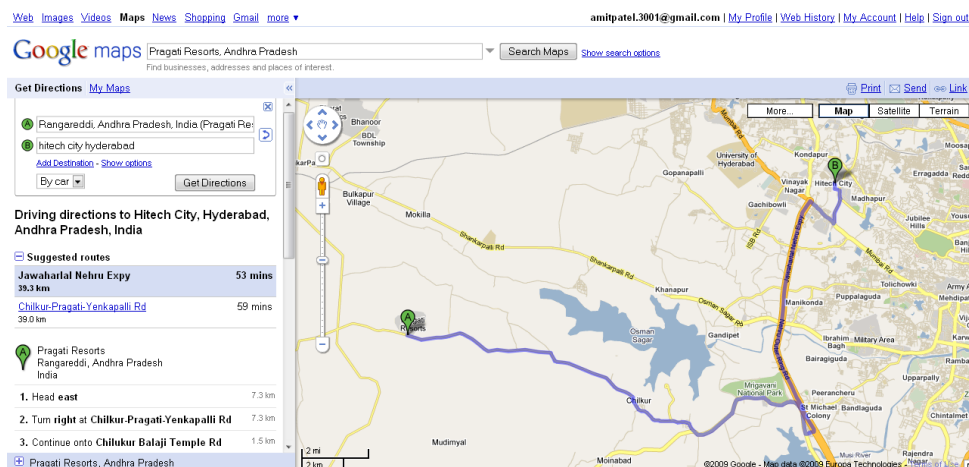
Ταυτότητα

Όπου κι' αν τρέχει μια εφαρμογή, τοπικά ή στο «νέφος», θα πρέπει να υπάρχουν χρήστες οι οποίοι θα έχουν μια συγκεκριμένη εργασία. Αυτό επιτυγχάνεται με μια ψηφιακή ταυτότητα όπου προσδιορίζει ποιος και πού είναι ο χρήστης και τι επιτρέπεται να κάνει. Στο «νέφος» υπάρχουν συγκεκριμένες υπηρεσίες ταυτότητας. Η Microsoft έχει το Windows Live ID, το App Engine της Google απαιτεί λογαριασμό της Google και η Amazon έχει την δική της υπηρεσία ταυτότητας.

Χαρτογράφηση

Οι χάρτες γίνονται όλο και πιο δημοφιλείς σε Web εφαρμογές. Οι περισσότεροι δικτυακοί τόποι δείχνουν τις θέσεις τους και δίνουν οδηγίες στους επισκέπτες.

Υπηρεσίες χαρτογράφησης «νέφος» παρέχουν η Google με το Google Earth και η Microsoft με το Virtual Earth.



Εικόνα 9. Η υπηρεσία Google Maps⁹

Αναζήτηση

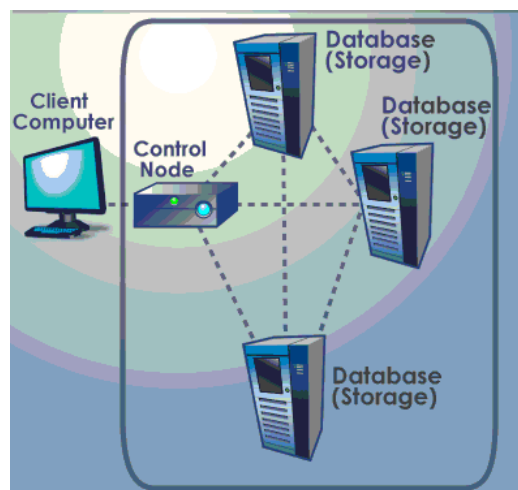
Δεν είναι κάτι νέο, αλλά είναι κάτι πολύ λειτουργικό. Δίνει την δυνατότητα αναζήτησης και επιστρέφει πίσω δυο τύπους αποτελεσμάτων αυτά που βρίσκονται στο δικό μας Web και αυτά που βρίσκονται σε όλο το Internet.

⁹ Εικόνα από <http://www.amitbhawani.com/blog/find-directions-between-two-locations/>

3. Αποθήκευση Στο «Νέφος»

Η αποθήκευση «νέφους» περιλαμβάνει την αποθήκευση δεδομένων σε έναν παροχέα υπηρεσιών «νέφους» αντί σε ένα τοπικό σύστημα. Έχει διάφορα πλεονεκτήματα σε σχέση με την παραδοσιακή αποθήκευση, αν τα δεδομένα είναι σε ένα «νέφος» τότε μπορούν να προσπελαστούν από οποιαδήποτε θέση στο Internet. Υπάρχουν πολλά διαφορετικά συστήματα αποθήκευσης στο «νέφος» και είναι συγκεκριμένα σ' αυτό που κάνουν.

Ένα σύστημα αποθήκευσης στο «νέφος» χρειάζεται έναν διακομιστή δεδομένων συνδεδεμένο με το Internet. Αν ένας Υπολογιστής-Πελάτης θέλει να ανακτήσει δεδομένα αποκτά πρόσβαση στον διακομιστή μέσω Web και ο διακομιστής του στέλνει τα αρχεία ή τον αφήνει να έχει πρόσβαση σ' αυτά.



Εικόνα 10. Ο Control Node διαχειρίζεται τα διασκορπισμένα δεδομένα¹⁰

Γενικά τα συστήματα αποθήκευσης στο «νέφος» χρησιμοποιούν δεκάδες ή εκατοντάδες διακομιστές δεδομένων παρέχοντας πλεονασμό. Χωρίς τον πλεονασμό τα συστήματα αποθήκευσης δεν θα μπορούσαν να εξασφαλίσουν ότι οι πελάτες θα μπορούν να έχουν πρόσβαση στα δεδομένα τους οποιαδήποτε στιγμή. Πολλοί χρησιμοποιούν αποθήκευση στο «νέφος» όχι γιατί δεν έχουν χώρο τοπικά, αλλά για λόγους ασφαλείας, αν κάτι συμβεί στο κτίριο τους δεν θα χάσουν τα δεδομένα τους.

¹⁰ Εικόνα από <http://computer.howstuffworks.com/cloud-computing/cloud-storage.htm>

Αποθήκευση ως Υπηρεσία

Αποθήκευση ως Υπηρεσία (SaaS) σημαίνει ότι ένας τρίτος προμηθευτής μισθώνει αποθηκευτικό χώρο σε τελικούς χρήστες οι οποίοι δεν έχουν το κεφάλαιο να τον αγοράσουν. Αυτή η τακτική δεν είναι κάτι καινούριο, αλλά λόγω των αντιγράφων ασφαλείας και την ανάγκη αποκατάστασης από καταστροφές έχει γίνει δημοφιλείς. Το μεγαλύτερο πλεονέκτημα είναι η εξοικονόμηση κόστους. Ο παροχέας χρησιμοποιεί μια μονάδα κόστους ανά gigabyte χώρου ή ανά δεδομένα που μεταφέρονται. Το θετικό είναι ότι ο τελικός χρήστης δεν πληρώνει για την υποδομή αλλά για τα δεδομένα που μεταφέρει.

Πάροχοι

Υπάρχουν πολλοί παροχείς αποθήκευσης στο «νέφος» και αυξάνονται κάθε μέρα. Μερικά παραδείγματα ειδικευμένων παροχέων «νέφους» είναι:

- Τα [Google Docs](#) επιτρέπουν στους χρήστες να στέλνουν έγγραφα, λογιστικά φύλλα και παρουσιάσεις σε διακομιστές δεδομένων του Google και στην συνέχεια μπορούν να επεξεργαστούν με μια Google εφαρμογή.
- Οι παροχείς ηλεκτρονικού ταχυδρομείου όπως οι [Gmail](#), [Hotmail](#) και [Yahoo! Mail](#) αποθηκεύουν τα μηνύματα των χρηστών σε δικούς τους διακομιστές.
- Τα [Flickr](#) και [Picasa](#) φιλοξενούν εκατομμύρια ψηφιακές φωτογραφίες. Οι χρήστες μπορούν να δημιουργήσουν τα δικά τους online λευκώματα.
- Το [YouTube](#) φιλοξενεί εκατομμύρια βίντεο που στέλνουν οι χρήστες.
- Το [Facebook](#) και το [MySpace](#) είναι κοινωνικοί δικτυακοί τόποι δικτύωσης, τα μέλη των οποίων μπορούν να δημοσιεύουν φωτογραφίες και άλλα περιεχόμενα.

Πολλές από αυτές τις υπηρεσίες προσφέρονται δωρεάν, αλλά χρεώνονται ανάλογα με τα αποθηκευμένα gigabyte. Όσο πέφτουν οι τιμές οι επιχειρήσεις προσφέρουν ορισμένο χώρο δωρεάν.

Ασφάλεια

Για την ασφάλιση των δεδομένων χρησιμοποιούνται κάποιες τεχνικές:

- **Κρυπτογράφηση:** Ένας πολύπλοκος αλγόριθμος χρησιμοποιείται για να κωδικοποιήσει τις πληροφορίες. Ο χρήστης για να τις αποκωδικοποιήσει χρειάζεται το κλειδί κρυπτογράφησης. Ενώ είναι εύκολο να σπάσει κάποιος τις κρυπτογραφημένες πληροφορίες χρειάζεται μεγάλη υπολογιστική ισχύ.
- **Διαδικασίες ελέγχου ταυτότητας:** Ο χρήστης δημιουργεί ένα όνομα και έναν κωδικό πρόσβασης.
- **Πρακτικές εξουσιοδότησης:** Ο χρήστης ορίζει έναν αριθμό ανθρώπων οι οποίοι είναι εξουσιοδοτημένοι να έχουν πρόσβαση στις πληροφορίες. Η εξουσιοδότηση έχει κλιμάκια, για παράδειγμα ένας υπάλληλος έχει περιορισμένη πρόσβαση σε σχέση με τον προϊστάμενο του.

Και πάλι όμως υπάρχει η ανησυχία ότι τα δεδομένα είναι τρωτά, και αυτό συμβαίνει διότι κάποιος τρίτος τα διαχειρίζεται.

Πλεονεκτήματα

Η επιλογή αποθήκευσης στο «νέφος» γίνεται όλο και πιο ελκυστική γιατί τα δεδομένα αποθηκεύονται στο Web διασκορπισμένα παρά όλα μαζί τοπικά. Οι παροχείς ισορροπούν τον φόρτο στους διακομιστές μεταφέροντας τα δεδομένα μεταξύ διάφορων κέντρων δεδομένων εξασφαλίζοντας ότι είναι αποθηκευμένα κοντά για να είναι γρήγορα διαθέσιμα. Συμφέρει γιατί τα δεδομένα προστατεύονται σε περίπτωση κάποιας καταστροφής.

Κλοπή

Επειδή τίποτα στον κόσμο δεν είναι τέλει, όταν τα δεδομένα είναι αποθηκευμένα απομακρυσμένα υπάρχει πιθανότητα κλοπής ή μη εξουσιοδοτημένης παρακολούθησης, προκαλώντας έτσι προβλήματα ασφαλείας. Θα μπορούσαν τα δεδομένα μιας επιχείρησης να είναι αποθηκευμένα δίπλα σ' αυτά ενός ανταγωνιστή και ο κίνδυνος να δει τις πληροφορίες είναι πραγματικός. Αυτό θα μπορέσει να αποφευχθεί μόνο με κρυπτογράφηση των δεδομένων και ασφάλιση στην μεταφορά δεδομένων.

4. Το Cloud Computing στις επιχειρήσεις

Το Cloud Computing δεν μπορεί να ανταποκριθεί σε όλες τις ανάγκες μιας επιχείρησης. Όπως το υλικό και το λογισμικό είναι διαφορετικό ανάμεσα σε δυο επιχειρήσεις έτσι και οι ανάγκες τους σε Cloud Computing είναι διαφορετικές.

Πότε πρέπει να χρησιμοποιηθεί το Cloud Computing

Αν μια επιχείρηση πρέπει να στραφεί σε λύση Cloud Computing εξαρτάται από διάφορους παράγοντες:

- Την αναλογία κόστους/κέρδους
- Την ταχύτητα παράδοσης
- Το μέγεθος αποθηκευτικού χώρου
- Την δομή της επιχείρησης

Μερικές φορές το Cloud Computing ταιριάζει τέλεια σε μια επιχείρηση, άλλες φορές όμως μπορεί να είναι λιγότερο αποδοτικό.

Υπολογιστικά «σύννεφα»

Είναι κλιμακούμενοι, φθηνοί και κατ' απαίτηση υπολογιστικοί πόροι. Τρία παραδείγματα είναι:

- Το EC2 του Amazon
- Το Google App Engine
- Το Berkeley Open Infrastructure for Network Computing (BOINC)

Αυτές οι υπηρεσίες είναι καλές για επιχειρήσεις οποιουδήποτε μεγέθους αλλά οι μεγάλες επιχειρήσεις μπορεί να βρίσκονται σε μειονεκτική θέση επειδή αυτές οι υπηρεσίες δεν προσφέρουν την τυπική διαχείριση, παρακολούθηση και ικανότητες διακυβέρνησης που αυτές οι επιχειρήσεις έχουν συνηθίσει να χρησιμοποιούν. Αν και η Amazon προσφέρει υποστήριξη επιπέδου μεγάλης επιχείρησης όπως και άλλοι αναπτυσσόμενοι παροχείς.

Αποθήκευση στο «Νέφος»

Η αποθήκευση στο «νέφος» ήταν από τις πρώτες προσφορές του Cloud Computing στην αγορά. Ακόμα όμως μένει μια πολύ δημοφιλής λύση. Υπάρχουν εκατοντάδες προμηθευτές που προσφέρουν αποθήκευση στο «νέφος». Αυτή είναι η

ιδανική λύση αν μια επιχείρηση θέλει να έχει τα δεδομένα της εκτός εταιρείας. Η ασφάλεια και το κόστος είναι τα κύρια θέματα σ' αυτήν την επιλογή. Ο προμηθευτής παίζει μεγάλο ρόλο σε αυτό. Για παράδειγμα το S3 της Amazon είναι η κορυφαία λύση.

Εφαρμογές «νέφους»

Οι εφαρμογές «νέφους» διαφέρουν από το υπολογιστικό «νέφος» γιατί χρησιμοποιούν εφαρμογές λογισμικού βασισμένες στο Web και παραδίδονται στους χρήστες μέσω browser. Οι εφαρμογές αυτές διαφέρουν από τις κοινές εφαρμογές διότι δεν χρειάζονται εγκατάσταση σε τοπικούς υπολογιστές της επιχείρησης και έτσι μειώνεται το κόστος συντήρησης του λογισμικού. Μερικές εφαρμογές «νέφους» περιλαμβάνουν:

- Web εφαρμογές(MySpace, YouTube)
- SaaS (Google Apps)
- Λογισμικό συν υπηρεσίες(Microsoft online services)

Πλεονεκτήματα

Το Cloud Computing μπορεί να ωφελήσει διαφορετικά μια εταιρεία από μια άλλη, αλλά σε όλες τις περιπτώσεις βοηθά την μηχανογράφηση.

Κλιμάκωση

Για μια εταιρεία που χρειάζεται τεράστια επέκταση σε υπολογιστική ισχύ το Cloud Computing μπορεί να βοηθήσει. Αντί να αγοράσει και να εγκαταστήσει τον εξοπλισμό, μπορεί να αγοράσει πρόσθετους κύκλους CPU ή χώρο αποθήκευσης από έναν τρίτο. Όταν θα έρθει η στιγμή να μην χρειάζεται αυτόν τον εξοπλισμό, μπορεί να σταματήσει να χρησιμοποιεί τις υπηρεσίες του προμηθευτή. Η ιδέα είναι ότι προστίθεται εξοπλισμός μόνο όταν χρειάζεται και αφαιρείται όταν δεν χρειάζεται.

Περισσότεροι Εσωτερικοί Πόροι

Μεταφέροντας τα περισσότερα δεδομένα σε κάποιον τρίτο, το τμήμα μηχανογράφησης της εταιρείας ελευθερώνεται να δουλέψει σε σημαντικές τοπικές εργασίες. Επίσης δεν χρειάζεται εργατικό δυναμικό για την μηχανογράφηση. Οι διακοπές λειτουργίας είναι ο εφιάλτης του προσωπικού μηχανογράφησης αλλά αυτό το φορτίο μεταφέρεται πλέον στον παροχέα. Γι' αυτό ο παροχέας πρέπει να προσφέρει υποστήριξη εικοσιτέσσερις ώρες το εικοσιτετράωρο.

Ασφάλεια

Υπάρχουν πολλοί κίνδυνοι για μια εταιρεία όταν χρησιμοποιεί το «νέφος» αλλά οι μεγάλες επιχειρήσεις στο Cloud Computing προσπαθούν να τους ελαττώσουν. Οι προμηθευτές έχουν αυστηρά μέτρα ασφαλείας όπως αποδεδειγμένες κρυπτογραφικές μεθόδους για επικύρωση χρηστών. Μια εταιρεία μπορεί να κρυπτογραφήσει η ίδια τα δεδομένα της πριν τα στείλει στο «νέφος». Με τόσα μέτρα τα δεδομένα είναι πιο ασφαλή στο «νέφος» παρά μέσα στην ίδια την επιχείρηση.

Μειονεκτήματα

Υπάρχουν πολλές περιπτώσεις όπου το Cloud Computing δεν αποτελεί την καλύτερη λύση. Επίσης μερικές εφαρμογές δουλεύουν καλύτερα τοπικά.

Οι ευαίσθητες πληροφορίες

Όταν τα δεδομένα αποθηκεύονται σε κάποιον τρίτο τότε χάνεται ένα επίπεδο ελέγχου. Έχει αποδειχτεί ότι οι κυβερνητικοί ανακριτές είναι πιο εύκολο να πάρουν πληροφορίες από τρίτους παρά από ιδιόκτητο διακομιστή. Επίσης οι παροχείς μπορούν να μοιραστούν δεδομένα με διαφημιστικές εταιρείες.

Αυτό βέβαια δεν σημαίνει ότι μια επιχείρηση δεν μπορεί να έχει τα δεδομένα της σ' ένα «νέφος», θα πρέπει απλώς να είναι ασφαλείς. Ο καλύτερος τρόπος είναι η κρυπτογράφηση πριν σταλούν τα δεδομένα στον παροχέα γιατί μόνο όποιος έχει το κλειδί αποκρυπτογράφησης μπορεί να δει τα δεδομένα. Αλλά κάποια δεδομένα που προκύπτουν από online εφαρμογές δεν μπορούν να κρυπτογραφηθούν.

Εφαρμογές που δεν υποστηρίζονται

Μερικές εφαρμογές δεν είναι έτοιμες να χρησιμοποιηθούν στο Cloud Computing λόγω μικρών προβλημάτων ή μη λειτουργίας. Ένας λόγος είναι μια εφαρμογή να χρησιμοποιεί μεγάλο εύρος ζώνης για να επικοινωνεί με τους χρήστες ή να είναι δύσκολο να ενοποιηθεί με άλλες εφαρμογές. Σ' αυτή την περίπτωση είναι πιο οικονομικό να τρέχει τοπικά η εφαρμογή. Επίσης εάν η εφαρμογή συνεργάζεται με μια βάση δεδομένων που βρίσκεται τοπικά τότε είναι αναγκασμένη να τρέξει κι' αυτή τοπικά. Άλλος λόγος είναι ότι μερικές εφαρμογές δεν μπορούν να επικοινωνούν με ασφάλεια στο Internet. Όλα αυτά δεν χρειάζεται να προκαλούν απογοήτευση, μπορεί πάντα κάποιος να δημιουργήσει τις δικές του εφαρμογές.

Θέματα ασφαλείας

Η ασφάλεια στο Cloud Computing είναι δίκοπο μαχαίρι, υπάρχουν αρνητικά και θετικά. Σε έρευνα που πραγματοποιήθηκε ανάμεσα σε 244 ανώτερους υπαλλήλους μηχανογράφησης, η μεγαλύτερη ανησυχία πάνω στο Cloud Computing ήταν η ασφάλεια.

Μυστικότητα

Το μεγαλύτερο και πιο εμφανές θέμα είναι η μυστικότητα. Από την στιγμή που όλα τα δεδομένα μια επιχείρησης βρίσκονται σε κάποιον τρίτο υπάρχει ανησυχία. Υπάρχουν όμως προμηθευτές που ως σκοπό έχουν την ασφάλεια των δεδομένων που φιλοξενούν και κάνουν τα πάντα για να κρατούν τους πελάτες του ευτυχισμένους. Υπάρχει πάντα όμως η περίπτωση εισβολής, σε αυτή την περίπτωση η καλύτερη λύση είναι στις κρίσιμες εργασίες να γίνονται εκτενείς ελέγχοι ασφαλείας.

Οφέλη

Κατ' αρχήν η συγκέντρωση των δεδομένων είναι ένα πολύ θετικό στοιχείο. Περισσότεροι από 12000 υπολογιστές χάνονται στα Αμερικάνικα αεροδρόμια κάθε χρόνο. Έτσι εάν τα δεδομένα είναι διασκορπισμένα είναι πιο εύκολο να χαθούν. Με αυτόν τον τρόπο μπορεί να γίνει πιο εύκολη και η παρακολούθηση τους.

Επίσης σε περίπτωση παραβίασης τα δεδομένα μεταφέρονται εύκολα σε άλλον υπολογιστή και με ομαλό τρόπο για τους χρήστες που τα χρησιμοποιούν εκείνη την στιγμή.

Νομικά θέματα

Επειδή το Cloud Computing βρίσκεται σε μεταβατικό στάδιο ακόμα, δεν υπάρχουν κάποιοι νόμοι σε ισχύ. Έτσι μερικοί παροχείς είτε δεν προσφέρουν ασφάλεια είτε χρησιμοποιούν τα δεδομένα που φιλοξενούν.

Κανονισμοί

Αυτήν τη στιγμή δεν υπάρχει κανένας ισχύων κανονισμός αλλά για να αναπτυχθεί το Cloud Computing πρέπει να υπάρξει. Για παράδειγμα όταν χρεοκόπησε η Washington Mutual εμφανίστηκε ως η μεγαλύτερη χρεοκοπία τράπεζας στην ιστορία. Επειδή υπήρχαν κανονισμοί όμως η κυβέρνηση των ΗΠΑ ξεπλήρωσε όλους τους πελάτες της. Στο Cloud Computing από την στιγμή που δεν υπάρχει κανένας κανονισμός κανένας δεν εγγυάται εάν ένας παροχέας αποφασίσει να κλείσει και χαθούν τα δεδομένα.

5. Το Μέλλον του Cloud Computing

Όπως είπαμε είμαστε στα πρώτα στάδια του Cloud Computing όπου πολλές οργανώσεις κάνουν τα πρώτα δειλά βήματα τους, αλλά μέχρι το 2020 το «νέφος» προβλέπεται να είναι ένα μεγάλο και μόνιμο κομμάτι της τεχνολογίας των υπολογιστών. Προβλέπεται να δούμε μείωση της ισχύς των επεξεργαστών, μείωση του φόρτου εργασίας και αυτοματοποιημένα κέντρα δεδομένων με επεκτάσιμη αρχιτεκτονική.

Η αξία του Cloud Computing στην αγορά, αναμένεται να αυξηθεί από 35 δις δολάρια το 2011 σε 150 δις δολάρια το 2020. Παράλληλα με την αύξηση της ζήτησης από τις επιχειρήσεις θα υπάρξει ανάπτυξη στις τεχνολογίες που υποστηρίζουν τα «σύννεφα» με ταχεία αύξηση της επεξεργαστικής ισχύος κάνοντας τα ακόμα φθηνότερα.

Σε όσες έρευνες έχουν γίνει τα αποτελέσματα προκαλούν έκπληξη καθώς το μεγαλύτερο ποσοστό είναι θετικό όσον αφορά το Cloud Computing ενώ βρίσκεται ακόμα στην αρχή. Γιατί οι κυβερνήσεις υιοθετούν υπηρεσίες Cloud Computing σε υψηλότερο από αναμενόμενο ποσοστό;

Το «νέφος» μπαίνει στο επόμενο στάδιο της ανάπτυξης του και αυτό που κάνει είναι να ελαττώνει να μειονεκτήματα του και να αυξάνει τα πλεονεκτήματα του. Ετοιμαζόμαστε λοιπόν να το υποδεχτούμε και να βρούμε όσους περισσότερους τρόπους μπορούμε να ωφεληθούμε από αυτό.

3. Εισαγωγή στο OpenStack

Το OpenStack αποτελεί μια συλλογή λογισμικού ανοικτού κώδικα, η οποία δίνει την δυνατότητα εγκατάστασης και χρήσης της τεχνολογίας cloud computing. Η Rackspace με την πλατφόρμα της 'Cloud Files' και η NASA με το 'Nebula', αποτελούν τους αρχικούς συνεισφέροντες του OpenStack. Πλέον υπάρχουν πάνω από 150 μέλη, συμπεριλαμβανομένων της Canonical, Dell και Citrix. Το OpenStack παρέχει τρεις υπηρεσίες cloud computing : Υπολογιστική Ισχύ (Compute), Υποδομή Αποθήκευσης (Object Storage) και Υπηρεσία Εικόνων(Image Service). Στην τρέχων διανομή (Essex) έχουν προστεθεί άλλες δύο υπηρεσίες που παρέχουν υποδομές για την υποστήριξη των υπολοίπων. Ταμπλό διαχείρισης (Dashboard) και Ταυτοποίηση (Identity).

Κωδικά ονόματα Υπηρεσιών :

- Compute - Nova
- Object Storage - Swift
- Image - Glance
- Dashboard - Keystone
- Identity – Horizon

1. Υπηρεσία Υπολογιστικής Ισχύος – Nova

Το Nova αποτελεί πλατφόρμα διαχείρισης του Cloud. Χειρίζεται όλες τις απαραίτητες ενέργειες του κύκλου ζωής των instances¹¹ στο cloud. Ειδικότερα, το Nova είναι υπεύθυνο, για την παροχή εικονικών μηχανών, ρύθμιση δικτύου των instances, και της κλιμακούμενης επεκτασιμότητας της εγκατάστασης του cloud ανάλογα με το

¹¹ Η Amazon εισήγαγε την έννοια του 'instance' και αποτελεί την εικονική μηχανή που δημιουργεί(create), τρέχει(launch) και τερματίζει(terminate) ένας χρήστης ,μέσω φόρτωσης εικόνων λογισμικού που παρέχει η υπηρεσία.

αριθμό των servers. Τέλος, το Nova εκθέτει όλες τις δυνατότητές του μέσω API υπηρεσιών ιστού που είναι παρόμοιο με το EC2¹² και το EBS¹³ της Amazon.

Συνθετικά μέρη του Nova

- API Server (nova-api)
- Message Queue (rabbit-mq server)
- Compute Workers (nova-compute)
- Network Controller (nova-network)
- Volume Worker (nova-volume)
- Scheduler (nova-scheduler)

API Server (nova-api)

Ο API Server παρέχει μια διεπαφή για να αλληλεπιδρά και να διαχειρίζεται ο έξω κόσμος την υποδομή του νέφους. Η διαχείριση γίνεται μέσω αιτήσεων των διαδικτυακών υπηρεσιών χρησιμοποιώντας το EC2 API, και στη συνέχεια ο Server API, επικοινωνεί με τα σχετικά τμήματα της υποδομής του cloud μέσω του Message Queue. Τέλος, ως εναλλακτική λύση για το EC2 API, η OpenStack παρέχει δικό της API που ονομάζεται "OpenStack API".

Message Queue (rabbit-mq server)

Τα διάφορα τμήματα της OpenStack επικοινωνούν μεταξύ τους δια μέσου του πρωτοκόλλου AMQP (Advanced Message Queue Protocol). Το Nova χρησιμοποιεί ασύγχρονη μέθοδο επικοινωνίας για τις αιτήσεις και τις απαντήσεις, γεγονός που εξασφαλίζει ότι καμία ενέργεια χρήστη θα βρίσκεται σε αναμονή για μεγάλο χρονικό διάστημα.

Compute Workers (nova-compute)

Οι Compute Workers, ασχολούνται με την διαχείριση των του κύκλου ζωής των instances. Αποτελούν δαίμονες του συστήματος και λαμβάνουν αιτήματα μέσω του Message Queue και τα εκτελούν. Υπάρχουν αρκετοί Compute Workers σε μια τυπική εγκατάσταση Cloud, για μεγαλύτερη αποδοτικότητα του συστήματος.

¹² Το EC2 (Amazon Elastic Compute Cloud), επιτρέπει στους χρήστες να 'νοικιάζουν' εικονικούς υπολογιστές στους οποίους εκτελούν τις εφαρμογές τους, με άλλα λόγια το EC2 είναι υπεύθυνο για την λειτουργία των instances.

¹³ Το EBS (Elastic Block Storage), παρέχει αποθηκευτικό για τα instances του EC2, με την δέσμευση και επισύναψη τόμων. Τα δεδομένα παραμένουν και μετά τον τερματισμό και υποστηρίζει διαμόρφωση σε κάποιο σύστημα αρχείων.

Network Controller (nova-network)

Ο Network Controller είναι υπεύθυνος για την ρύθμιση του δικτύου των μηχανών υπηρεσίας (host machines) και συγκεκριμένα για την διευθυνσιοδότηση των IP και τη διαμόρφωση των VLANs¹⁴ και του δικτύου των κόμβων (compute nodes¹⁵)

Volume Worker (nova-volume)

Οι

Volume Worker χρησιμοποιούνται για την διαχείριση των LVM-based¹⁶ τόμων (volumes¹⁷) των instances. Ειδικότερα ,εκτελούν ενέργειες όπως δημιουργία , διαγραφή ,σύναψη και απόσπαση τόμων των instances. Το σημαντικότερο κομμάτι είναι, η σταθερή παροχή αποθηκευτικού χώρου ακόμη και όταν ένα instance τερματίζεται ή αποσπάται κάποιος τόμος του από τον χρήστη. Αυτό σημαίνει ότι τα δεδομένα του τόμου, είναι διαθέσιμα με την επανα-σύναψη του στο ίδιο ή και σε άλλα instances.

Scheduler (nova-scheduler)

Ο scheduler 'χαρτογραφεί' τις κλήσεις από και προς τα κατάλληλα τμήματα του OpenStack. Τρέχει ως δαίμονας¹⁸ (daemon) συστήματος και λαμβάνει αιτήσεις υπηρεσιών από τα instances και τις δρομολογεί στους compute servers που πρέπει να τρέξουν ,ανάλογα με έναν αλγόριθμό χρονοπρογραμματισμού. Οι βασικότεροι αλγόριθμοι είναι :

Chance : τυχαία επιλογή από διαθέσιμους πόρους

Availability zone : τυχαία επιλογή μέσα από ειδική λίστα διαθέσιμων πόρων

Simple : επιλέγεται διακομιστής με το μικρότερο φόρτο εργασίας

¹⁴ Το VLAN (εικονικό δίκτυο), είναι ένα δίκτυο υπολογιστών που αποτελείται από εικονικές συνδέσεις. Η εικονική σύνδεση χρησιμοποιεί μεθόδους εικονικής δικτύωσης.

¹⁵ Compute nodes, αποτελούν τους κόμβους δικτύου για την παροχή των instances .

¹⁶ Το LVM(Logical Volume Manager) παρέχει μεθόδους διαχείρισης και κατανομής χώρου των συσκευών μαζικής αποθήκευσης δεδομένων.

¹⁷ Με τον όρο Volume, περιγράφεται μια προσβάσιμη περιοχή αποθήκευσης από ένα Λειτουργικό Σύστημα, διαμορφωμένη με κάποιο σύστημα αρχείων(file system).

¹⁸ Ο δαίμονας είναι μια διεργασία η οποία εκτελείται στο παρασκήνιο.

2. Υπηρεσία Εικόνων - Glance

Το Glance αποτελεί το τμήμα του OpenStack που είναι υπεύθυνο για τις εικόνες εικονικών μηχανών. Η αποθήκευση εικόνων γίνεται με την υπηρεσία S3 της Amazon, για την απλή αποθήκευση και ανάκτηση δεδομένων στο web.

Συνθετικά μέρη του Glance

- Glance-api
- Glance-registry

Glance-api

Δέχεται αιτήματα για την αποθήκευση και ανάκτηση αρχείων εικόνων.

Glance-registry

Αποθηκεύει διεργασίες και ανακτά μεταδεδομένα των εικόνων. Παρέχει την δυνατότητα ρυθμίσεων της registry¹⁹ των διακομιστών.

3. Υποδομή Αποθήκευσης – Swift

Το Swift παρέχει ένα κατανεμημένο και συνεπές εικονικό μέσο αποθήκευσης. Είναι ικανό να φιλοξενεί τεράστιο όγκο δεδομένων που κατανέμονται κατά μήκος των κόμβων. Το Swift έχει ενσωματωμένο μηχανισμό διαχείριση αποτυχιών συστήματος και είναι σε θέση να αρχειοθετεί δεδομένα και να επεκτείνει το μέγεθος της χωρητικότητας του.

¹⁹ Η αποθήκευση ρυθμίσεων συστήματος και εφαρμογών σε ένα ενιαίο χώρο με μορφή βάσης δεδομένων ονομάζεται Registry.

Συνθετικά μέρη του Swift

- Swift Account
- Swift Container
- Swift Object
- Swift Proxy
- The RING

Swift Container server

Καταχωρεί σε λίστες τα αντικείμενα που είναι αποθηκευμένα, με την μορφή SQLite αρχείων. Επίσης παρέχει πληροφορίες για τον αριθμό των αντικειμένων και του χώρου που χρησιμοποιούν. Ο χώρος που αποθηκεύονται αυτά τα αντικείμενα ονομάζεται Container.

Swift Account server

Καταχωρεί σε λίστες τα Container

Swift Proxy server

Λειτουργεί ως 'φύλακας πύλης' του Swift ,και λαμβάνει τα αιτήματα από τον 'έξω' κόσμο, και στην συνέχεια τα δρομολογεί στους κατάλληλους παραλήπτες. Είναι υπεύθυνο για την αυθεντικοποίηση , του χρήστη που στέλνει τις αιτήσεις, και την εξουσιοδότηση του για την πρόσβαση στους πόρους που απαιτεί. Η εξουσιοδότηση γίνεται δια μέσου του Keystone. Επίσης διαχειρίζεται τις αποτυχίες με επανα-δρομολόγηση αιτημάτων.

Swift Object server

Είναι υπεύθυνο να χειρίζεται την αποθήκευση ,ανάκτηση και διαγραφή των αντικειμένων που βρίσκονται στον τοπικό χώρο αποθήκευσης. Τα αντικείμενα (Objects) βρίσκονται στο σύστημα αρχείων και είναι δυαδικά αρχεία που περιέχουν μεταδεδομένα²⁰ (metadata) ως επιπλέον χαρακτηριστικά αρχείων (xattr).

The RING

Παρέχει πληροφορίες για την φυσική διεύθυνση των αντικειμένων(objects) μέσα στο swift. Αναπαριστά την 'χαρτογράφηση' των ονομάτων των οντοτήτων στις φυσικές διευθύνσεις τους. Τέλος, ορισμένα τμήματα του Swift (Account, Container, Object) έχουν το δικό τους Ring.

²⁰ Ένα σύνολο μεταδεδομένων περιγράφει ένα πηγαίο σύνολο δεδομένων.

4. Υπηρεσία ταυτοποίησης – Keystone

Το Keystone είναι υπεύθυνο για την ταυτοποίηση και την παροχή πρόσβασης για όλα τα συστατικά στην οικογένεια OpenStack. Υλοποιεί το δικό του API (Identity API) και παρέχει επαλήθευση ταυτότητας και εξουσιοδότηση. Ο έλεγχος ταυτότητας επαληθεύει ότι η αίτηση προέρχεται από τον σωστό χρήστη και στην συνέχεια η εξουσιοδότηση επαληθεύει αν ο χρήστης έχει πρόσβαση στις υπηρεσίες που αναζητά. Η αυθεντικοποίηση γίνεται με δύο τρόπους. Ο πρώτος είναι με το όνομα χρήστη και κωδικό πρόσβασης, και ο δεύτερος στηρίζεται στην έννοια ‘authentication token’. Ειδικότερα, με το όρο authentication token , token ή διακριτικό αυθεντικοποίησης , περιγράφεται ένα ηλεκτρονικό κλειδί που δίνεται σε έναν εξουσιοδοτημένο χρήστη για να διευκολύνει την απόδειξη της ηλεκτρονικής του ταυτότητας. Η διαχείριση των χρηστών γίνεται με βάση το tenant στο οποίο ανήκουν, και τα δικαιώματα που τους παρέχει ο admin.

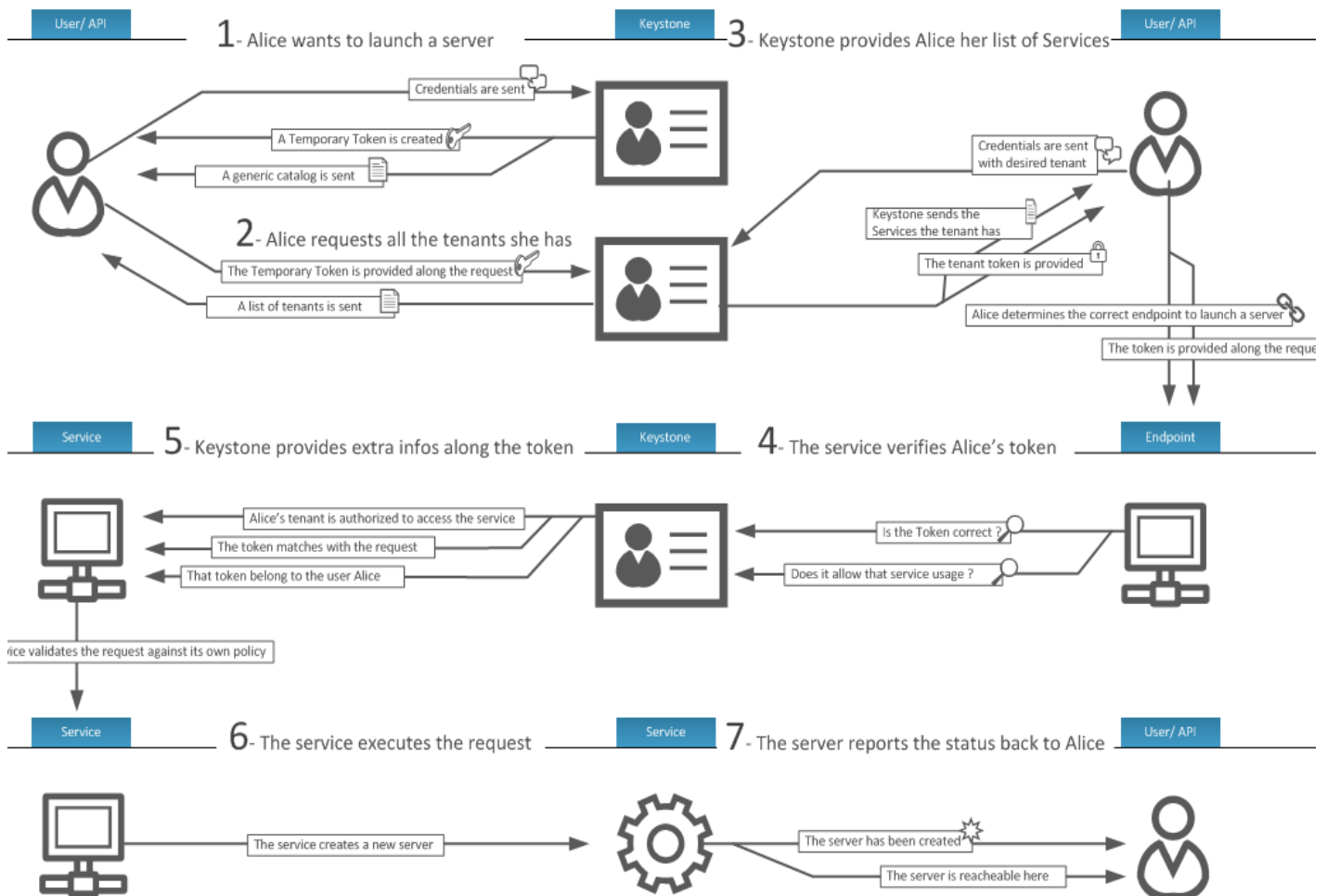
Βασικές έννοιες του Keystone

- **User** – η ψηφιακή αναπαράσταση ενός ατόμου , συστήματος ή υπηρεσίας που χρησιμοποιεί οποιαδήποτε υπηρεσία του OpenStack. Ένας χρήστης μπορεί να έχει διαφορετικούς ρόλους στο ίδιο ή και σε διαφορετικά tenants.
- **Credentials** – δεδομένα που ανήκουν και είναι γνωστά μόνο στον χρήστη (usernames, passwords ,tokens).
- **Endpoints** – αποτελούνται από την μοναδική διεύθυνση πόρου (URL) που τρέχει κάθε υπηρεσία των Nova, Swift και Glance και είναι προσβάσιμη από τον χρήστη.
- **Regions** – καθορίζει την φυσική τοποθεσία των ενεργών υπηρεσιών μέσα στην εγκατάσταση του cloud.
- **Services** – κάθε τμήμα που είναι συνδεδεμένο και εξουσιοδοτημένο μέσω του keystone ονομάζεται υπηρεσία. Κάθε υπηρεσία παρέχει endpoints δια μέσου των οποίων ο χρήστης έχει πρόσβαση στους πόρους του συστήματος (Nova, Swift, Glance).
- **Role** – ο ρόλος που έχει ένας χρήστης στο cloud , δηλαδή τα δικαιώματα και τα προνόμια του στο σύνολο των ενεργειών του.

- **Tenant** – είναι ένα σύνολο πληροφοριών των πόρων υπηρεσιών(endpoints) και των ρόλων που προσάπτονται στον χρήστη. Σε οποιαδήποτε κλήση προς τις υπηρεσίες του OpenStack πρέπει να καθοριστεί και το tenant που ανήκει ο χρήστης. Πρακτικά, το tenant αποτελεί μία υψηλού επιπέδου ομαδοποίηση χρηστών εντός του Keystone, στο οποίο ανήκουν εικονικές μηχανές δια μέσου του Nova και containers δια μέσου του Swift.

Η αυθεντικοποίηση γίνεται σύμφωνα με το διάγραμμα[12]:

The Keystone Identity Manager



Ταυτοποίηση με το Keystone

1. Ο υποψήφιος χρήστης 'Alice' επιθυμεί να ξεκινήσει έναν διακομιστή
 - Στέλνοντας τις συστάσεις του (Credentials) στο Keystone προς ταυτοποίηση
 - Δημιουργείται ένα προσωρινό Token από το Keystone
2. Ο χρήστης αιτεί το Tenant στο οποίο ανήκει
 - Δημιουργείται προσωρινό Token κατά την αίτηση
3. Το Keystone παρέχει στον χρήστη την λίστα υπηρεσιών
 - Παρέχεται η λίστα με τις υπηρεσίες του Tenant
 - Δημιουργείται το Token για τον χρήστη
 - Ελέγχεται η πρόσβαση του χρήστη με τα endpoints υπηρεσιών
4. Το Keystone επαληθεύει το Token του χρήστη
5. Το Keystone στέλνει μηνύματα προς τις υπηρεσίες του Tenant
 - Εξουσιοδοτεί τον χρήστη για κάθε υπηρεσία
 - Πιστοποίηση του ζητούμενου Token χρήστη
6. Η υπηρεσία εκτελεί την αίτηση του χρήστη
 - Δημιουργείτε ένας διακομιστής
7. Η υπηρεσία απαντά στον χρήστη
 - Ο διακομιστής δημιουργήθηκε
 - Είναι προσβάσιμος στο Endpoint του.

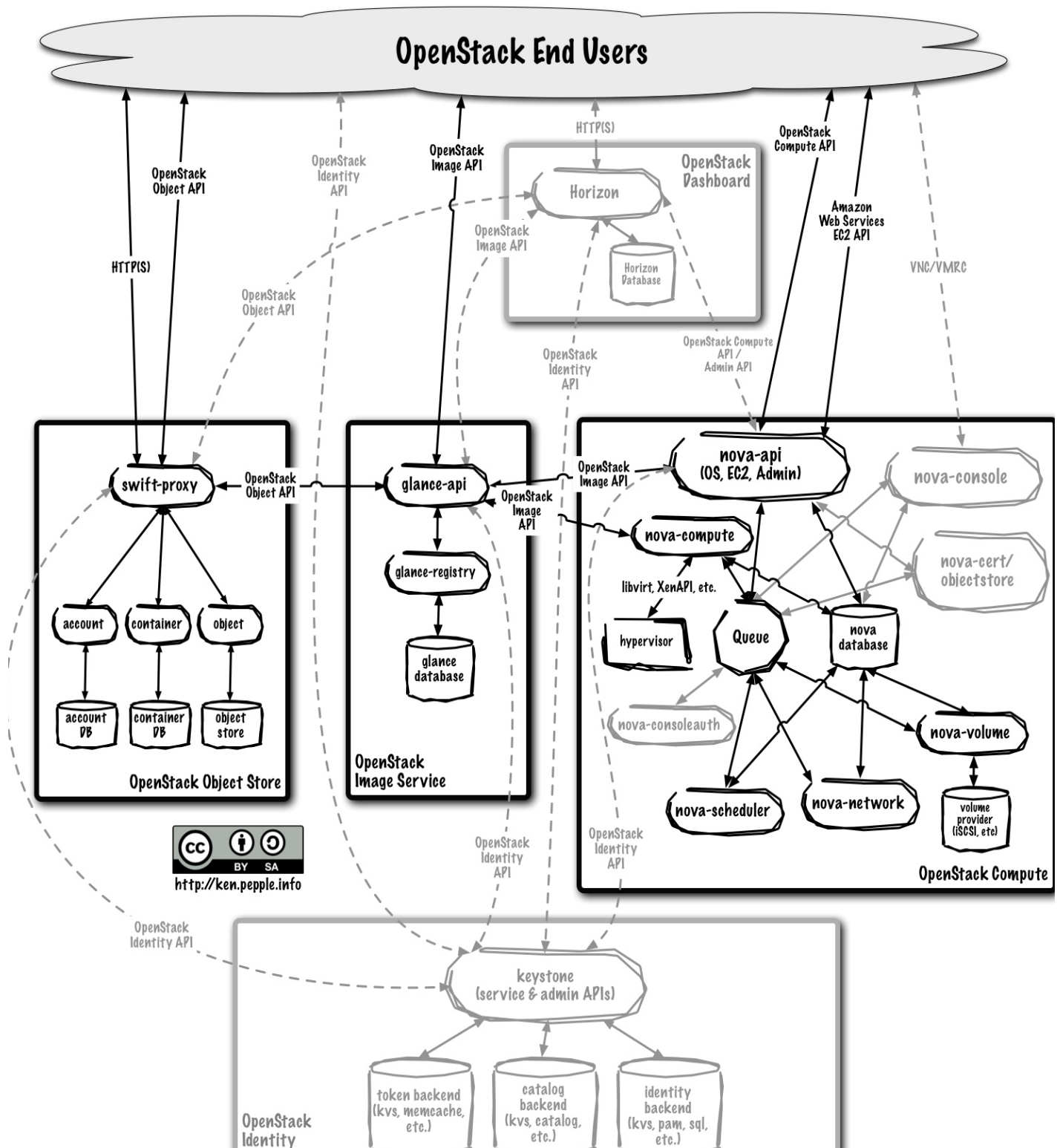
5. Γραφική διεπαφή διαχειριστή – Horizon

Το Horizon αποτελεί το web-based ταμπλό διαχείρισης των υπηρεσιών του OpenStack. Χρησιμοποιείται για τη διαχείριση των instances, των key pairs ,των χρηστών , των μονάδων(Volumes) του αποθηκευτικού χώρου των αντικειμένων και των εικόνων. Η χρήση του απευθύνεται στον διαχειριστή του Cloud, για μια εύκολη διερεύνηση της εγκατάστασης εκτός τερματικού. Αναλυτικότερα :

- **Διαχείριση των Instances** – Δημιουργία και τερματισμός ενός instance, παρέχει το ιστορικό κονσόλας, συνδέσεις VNC²¹ και επισύναψη μονάδων σε instance.
- **Διαχείριση χρηστών, εικόνων , μονάδων και χώρου αποθήκευσης** – Ενέργειες διαγραφής , δημιουργίας και επεξεργασίας.
- **Διαχείριση πρόσβασης και ασφάλειας** – Έλεγχος κλειδιών με Key pairs.
- **Διαχείριση των Flavors** – Το flavor αποτελεί μια προεπιλεγμένη διαθέσιμη διαμόρφωση του υλικού για ένα στιγμιότυπο της βάσης δεδομένων. Κάθε flavor έχει ένα μοναδικό συνδυασμό χαρακτηριστικών μνήμης και CPU.

²¹ VNC(Virtual Network Computing). Είναι ένα σύστημα το οποίο επιτρέπει την απομακρυσμένη διαχείριση επιφάνειας εργασίας ενός υπολογιστή, από οποιονδήποτε υπολογιστή στο διαδίκτυο..

6. Αρχιτεκτονική του OpenStack



Στο παραπάνω διάγραμμα [13], παρουσιάζεται η σύνδεση μεταξύ των υπηρεσιών του OpenStack και του τελικού χρήστη. Γίνεται εμφανής η διαφοροποίηση των κύριων υπηρεσιών νέφους (compute, storage και image services με έντονο περίγραμμα) με τις βοηθητικές υπηρεσίες του OpenStack(identity και dashboard με ελαφρό περίγραμμα). Αρχικά κάθε υπηρεσία διαθέτει δική της βάση δεδομένων και επικοινωνεί αμφίδρομα με κάποια άλλη, αλλά και με τον χρήστη μέσω ενός API υπηρεσίας (Διασύνδεση Προγραμματισμού Εφαρμογών), εκτός του Horizon, το οποίο χρησιμοποιεί HTTP προς τον χρήστη. Κάθε υπηρεσία συνδέεται στο Horizon έτσι ώστε να είναι δυνατή η προσπέλαση της μέσω του dashboard.

Παρόμοια με το Horizon, κάθε υπηρεσία επικοινωνεί με το Keystone έτσι ώστε να κάνει αιτήσεις και να λαμβάνει απαντήσεις που αφορούν την ταυτοποίηση και εξουσιοδότηση πρόσβασης του χρήστη τις επιθυμεί. Αποθηκεύει δεδομένα που αφορούν την εξουσιοδότηση χρήστη (token backend) , την λίστα με τις διαθέσιμες υπηρεσίες χρήστη (catalog backend) και την πρόσβαση σε υπηρεσίες (identity backend).

Τα συστατικά του Swift επικοινωνούν με τον έξω κόσμο δια μέσου του proxy server. Το οποίο δρομολογεί τα μηνύματα από και προς τις υπόλοιπες υπηρεσίες του Swift με την βοήθεια των πληροφοριών που προσφέρει το RING κάθε υπηρεσίας. Διαθέτει βάσεις δεδομένων για τον Object, Container και Account server.

Στο Glance η βάση δεδομένων παρέχει πληροφορίες στο glance-registry των εικόνων.

Μια σημαντική σημείωση είναι ότι μόνο μεταξύ του Swift και Nova δεν υπάρχει διασύνδεση. Το nova διαθέτει όλες τις υπηρεσίες του να συνεργάζονται και να ανταλλάσσουν πληροφορίες δια μέσου του Message Queue. Κάθε πληροφορία αποθηκεύεται στην βάση του nova, έτσι ώστε το nova-compute να μπορεί να διαχειρίζεται τον κύκλο ζωής των instances.

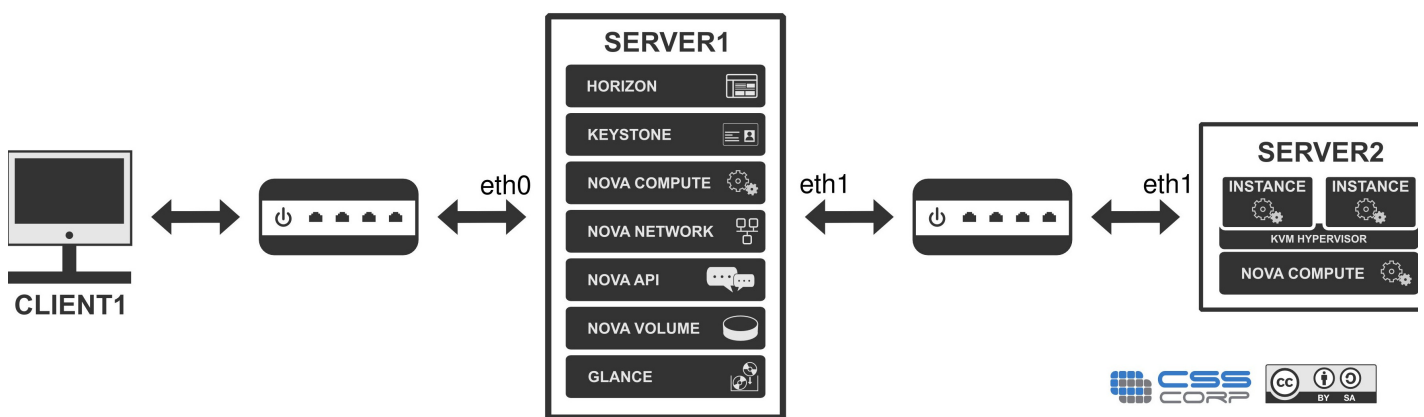
4. Εγκατάσταση του συστήματος

Εισαγωγή

Το παρόν κεφάλαιο περιγράφει την τυπική εγκατάσταση του OpenStack σε ένα μικρό δίκτυο cloud χρησιμοποιώντας τρεις μηχανές. Θα αναφερόμαστε σε αυτές τις μηχανές με τα κωδικά τους ονόματα Server1, Server2 και Client1.

Ο Server1 τρέχει όλες τις υπηρεσίες του OpenStack και λειτουργεί ως διαχειριστής του νέφους. Στον Server2 υπάρχει μόνο το τμήμα nova-compute της υπηρεσίας Nova, και ρόλος του είναι να τρέχει τα instances. Ο Client1 δεν έχει καμία υπηρεσία OpenStack, διαθέτει γραφικό περιβάλλον και χρησιμοποιεί στο bundling των εικόνων. Το bundling είναι η διαδικασία στην οποία δημιουργείτε ένα στιγμιότυπο μίας εικόνας μηχανής, και ενσωματώνονται συγκεκριμένες ρυθμίσεις και εφαρμογές που επιθυμεί ο χρήστης. Υπάρχουν και προετοιμασμένα στιγμιότυπα εικόνων διαθέσιμα για χρήση. Τέλος η δημιουργία εικόνων για το bundling προϋποθέτει ότι ο Client1 θα υποστηρίζει Τεχνολογία Εικονικοποίησης (VT)²².

Διάγραμμα συνδέσεων



²² VT(Virtual Technology) σε έναν υπολογιστή είναι δυνατότητα που του επιτρέπει να εκτελεί περισσότερα από ένα λειτουργικά συστήματα.

Πίνακας Εγκατάστασης

	Server1	Server2	Client1
Υπηρεσία	Cloud Controller	Compute Node	Bundling Εικόνων
Δίκτυο	eth0 – Δημόσιο eth1 – Ιδιωτικό	eth0 – Δημόσιο eth1 – Ιδιωτικό	eth0 – Δημόσιο
Διεύθυνση IP	eth0: 192.251.209.8 eth1: 192.168.0.1	eth0: 192.251.209.7 eth1: 192.168.0.2	eth0: 192.251.209.9
Hostname	it.uom.gr	it.uom.gr	it.uom.gr
Διακομιστής DNS	195.251.213.106 195.251.213.104	195.251.213.106 195.251.213.104	195.251.213.106 195.251.213.104
Gateway IP	195.251.209.1	195.251.209.1	195.251.209.1

1. Server1

Εγκατάσταση

Ο Server1 , όπως προαναφέρθηκε περιέχει όλα τα τμήματα του OpenStack. Το λειτουργικό σύστημα που τρέχει είναι η 64 bit έκδοση της διανομής Ubuntu server 12.04. Τα βασικά σημεία κατά την εγκατάσταση είναι η χειροκίνητη ρύθμιση των IP σύμφωνα με τις τιμές του Πίνακα εγκατάστασης, η επιλογή μόνο του πακέτου Openssh-server στο Packages menu. Στην διαμέριση δίσκου δημιουργούμε έναν primary 30GB ένα swap area 8GB και ρυθμίζουμε τον LVM(Δημιουργούμε ένα volume-group με το όνομα nova-storage).

Βασικές ενέργειες

Ενημέρωση του συστήματος

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install bridge-utils
```

Ρυθμίσεις Δικτύου

Επεξεργασία του αρχείου interfaces στην παρακάτω μορφή

```
nano /etc/network/interfaces
```


Προσθήκη:

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
    address 195.251.209.8
    netmask 255.255.255.0
    broadcast 195.251.209.255
    gateway 195.251.209.1
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 195.251.213.106 195.251.213.104

auto eth1
iface eth1 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
```

Επανεκκίνηση δικτύου

```
sudo /etc/init.d/networking restart
```

Διακομιστής NTP²³

Εγκατάσταση του πακέτου NTP , πρωτόκολλο δικτύου για να συγχρονιστούν αργότερα οι υπηρεσίες του OpenStack.

```
sudo apt-get install ntp
```

Επεξεργασία του αρχείου ntp.conf στην παρακάτω μορφή .

```
nano /etc/ntp.conf
```

προσθήκη:

```
server ntp.ubuntu.com
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

και επανεκκίνηση ntp

```
sudo service ntp restart
```

²³ NTP(Network Time Protocol) είναι ένα πρωτόκολλο δικτύωσης για συγχρονισμό των ρολογιών δύο ή περισσότερων υπολογιστικών συστημάτων.

Βάσεις Δεδομένων

Για τον διακομιστή της βάσης δεδομένων μπορεί να χρησιμοποιηθεί η MySQL , PostgreSQL ή SQLite, ανάλογα με την προτίμηση του διαχειριστή. Εμείς θα χρησιμοποιήσουμε MySQL.

Εγκατάσταση πακέτων MySQL.

```
sudo apt-get install mysql-server python-mysqldb
```

Αλλαγή διεύθυνσης bind από τον localhost (127.0.0.1) σε οποιοδήποτε host(0.0.0.0)

```
nano /etc/mysql/my.cnf
```

προσθήκη:

```
bind-address=0.0.0.0
```

επανεκκίνηση MySQL server

```
sudo restart mysql
```

Δημιουργία βάσεων δεδομένων για το nova , glance και keystone

δημιουργία βάσης με όνομα nova

```
sudo mysql -uroot -pmygreatsecret -e 'CREATE DATABASE nova;'
```

δημιουργία χρήστη novadbadmin

```
sudo mysql -uroot -pmygreatsecret -e 'CREATE USER novadbadmin;'
```

παροχή προνομίων στον χρήστη novadbadmin της βάσης nova

```
sudo mysql -uroot -pmygreatsecret -e "GRANT ALL PRIVILEGES ON nova.* TO 'novadbadmin'@'%' ;"
```

δημιουργία κωδικού πρόσβασης στον χρήστη novadbadmin

```
sudo mysql -uroot -pmygreatsecret -e "SET PASSWORD FOR 'novadbadmin'@'%' = PASSWORD('novasecret') ;"
```

δημιουργία βάσης με όνομα glance

```
sudo mysql -uroot -pmygreatsecret -e 'CREATE DATABASE glance;'
```

δημιουργία χρήστη glancedbadmin

```
sudo mysql -uroot -pmygreatsecret -e 'CREATE USER glancedbadmin;'
```

παροχή προνομίων στον χρήστη glancedbadmin της βάσης glance

```
sudo mysql -uroot -pmygreatsecret -e "GRANT ALL PRIVILEGES ON glance.* TO 'glancedbadmin'@'%' ;"
```

δημιουργία κωδικού πρόσβασης στον χρήστη glancedbadmin

```
sudo mysql -uroot -pmygreatsecret -e "SET PASSWORD FOR 'glancedbadmin'@'%' = PASSWORD('glancesecret') ;"
```

δημιουργία βάσης με όνομα keystone

```
sudo mysql -uroot -pmygreatsecret -e 'CREATE DATABASE keystone;'
```

δημιουργία χρήστη keystoneadmin

```
sudo mysql -uroot -pmygreatsecret -e 'CREATE USER keystoneadmin;'
```

παροχή προνομίων στον χρήστη keystoneadmin της βάσης keystone

```
sudo mysql -uroot -pmygreatsecret -e "GRANT ALL PRIVILEGES ON keystone.* TO 'keystoneadmin'@'%';"
```

δημιουργία κωδικού πρόσβασης στον χρήστη keystoneadmin

```
sudo mysql -uroot -pmygreatsecret -e "SET PASSWORD FOR 'keystoneadmin'@'%' = PASSWORD('keystonesecret');"
```

Keystone

Η πρώτη υπηρεσία OpenStack που θα εγκατασταθεί είναι το Keystone. Αποτελείτε από δύο τμήματα :

διαχείριση χρηστών – βάση δεδομένων για τους χρήστες μαζί με πληροφορίες ταυτοποίησης , ρόλων και ενεργειών στο «νέφος». (Βασικές έννοιες user, role , tenant)

κατάλογος υπηρεσιών – παρέχει τον κατάλογο με τις διαθέσιμες υπηρεσίες μαζί με τα endpoints τους. (Βασικές έννοιες services , endpoints)

Εγκατάσταση του keystone.

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

Επεξεργασία του αρχείου keystone.conf, το token που θα χρησιμοποιηθεί ονομάζεται 'admin' και για την βάση δεδομένων του keystone θα χρησιμοποιηθεί η MySQL . Αναγκαία συνθήκη για την χρήση του API διαχείρισης χρηστών, tenants κλπ είναι η ύπαρξη token στον admin, το οποίο διαμοιράζονται κρυφά οι υπηρεσίες του OpenStack.

```
nano /etc/keystone/keystone.conf
```

προσθήκη:

```
admin_token=admin
connection=mysql://keystoneadmin:keystonesecret@10.10.10.2/keystone
[username:password@server mysql/όνομα βάσης δεδομένων ]
```

επανεκκίνηση του keystone

```
sudo service keystone restart
```

συγχρονισμός βάσης δεδομένων

```
sudo keystone-manage db_sync
```

πρόσθεση μεταβλητών στο αρχείο ~/.bashrc, είναι αναγκαίες για την διεκπεραίωση των παρακάτω ενεργειών

```
SERVICE_ENDPOINT="http://localhost:35357/v2.0"  
SERVICE_TOKEN=admin
```

διαφορετικά μπορεί να γίνουν export στο τερματικό

```
export SERVICE_ENDPOINT="http://localhost:35357/v2.0"  
export SERVICE_TOKEN=admin
```

Διαχείριση χρηστών

Για την διαχείριση των υπηρεσιών του OpenStack , θα πρέπει να δημιουργηθεί ένα tenant για όλες, και χρήστες για την κάθε μια. Για να μπορούν να επικυρώνουν ,αιτήσεις ενός πελάτη, προστίθεται ο ρόλος του admin σε κάθε χρήστη υπηρεσίας. Επίσης, δημιουργείτε και ένα tenant για τον διαχειριστή του συστήματος, και ένα άλλο για της υπηρεσίες. Μια 'κρυφή' λεπτομέρεια που αφορά την δημιουργία χρηστών για τις υπηρεσίες είναι ότι , δεν αποτελούν κάποιο απλό χρήστη στην υποδομή του cloud, αλλά αποτελούν τον διαχειριστή τους. Τα credentials των διαχειριστών υπηρεσιών είναι αναγκαία όχι μόνο για την διασύνδεση τους με το Keystone αλλά και για την λειτουργία τους , γεγονός που γίνεται αντιληπτό παρακάτω στην παραμετροποίηση των αρχείων ini του API της κάθε υπηρεσίας.

Θα δημιουργήσουν δύο tenants, ένα admin (του διαχειριστή) και ένα service(των υπηρεσιών) .

```
keystone tenant-create --name admin  
keystone tenant-create --name service
```

Δημιουργία χρηστών

Θα δημιουργηθούν πέντε χρήστες : admin , nova , glance, swift και ec2.

```
keystone user-create --name admin --pass admin --email admin@foobar.com  
keystone user-create --name nova --pass nova --email nova@foobar.com  
keystone user-create --name glance --pass glance --email glance@foobar.com  
keystone user-create --name swift --pass swift --email swift@foobar.com
```

Δημιουργία ρόλων

Ένας ρόλος τύπου admin και ένας τύπου μέλος.

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

Λίστα των tenants ,χρηστών και ρόλων

Οι παρακάτω τιμές των πεδίων id τις κάθε λίστας πρόκειται να χρησιμοποιηθούν στο επόμενο βήμα, οπότε πρέπει να αποθηκευτούν σε ένα αρχείο κειμένου.

λίστα tenants

```
root@compute:/# keystone tenant-list
```

id	name	enabled
01e2786023034cec9903b3214ecc4b1f	service	True
a9f786df73104d6da5fb7cecd6a239d4	admin	True

λίστα χρηστών

```
root@compute:/# keystone user-list
```

id	enabled	email	name
5c86cf8c51b74c0493afe321ca8bacfd	True	admin@foobar.com	admin
6aa3198a22514ca69e7823623ce28972	True	glance@foobar.com	glance
f693a37b48e24758acd57796bce13efe	True	swift@foobar.com	swift
ffb5576d8cc8456280aac03bc09b489d	True	nova@foobar.com	nova

λίστα ρόλων

```
root@compute:/# keystone role-list
```

id	name
1671579958f441c2b52813a4d6407d68	admin
931a77710b004d988b871c8171b5bacd	Member

Πρόσθεση ρόλων στους χρήστες των Tenants

θα χρησιμοποιηθούν τα id's των παραπάνω λιστών, σύμφωνα με την εντολή :

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

πρόσθεση ρόλου 'admin' στον χρήστη 'admin' του tenant 'admin'

```
keystone user-role-add --user 5c86cf8c51b74c0493afe321ca8bacfd --role 1671579958f441c2b52813a4d6407d68 --tenant_id a9f786df73104d6da5fb7cecd6a239d4
```

πρόσθεση ρόλου 'admin' στους χρήστες 'nova', 'glance' και 'swift' του tenant 'service'

```
keystone user-role-add --user ffb5576d8cc8456280aac03bc09b489d --role 1671579958f441c2b52813a4d6407d68 --tenant_id 01e2786023034cec9903b3214ecc4b1f
keystone user-role-add --user 6aa3198a22514ca69e7823623ce28972 --role 1671579958f441c2b52813a4d6407d68 --tenant_id 01e2786023034cec9903b3214ecc4b1f
keystone user-role-add --user f693a37b48e24758acd57796bce13efe --role 1671579958f441c2b52813a4d6407d68 --tenant_id 01e2786023034cec9903b3214ecc4b1f
```

Κατάλογος Υπηρεσιών

Αφού έχουν δημιουργηθεί τα tenants με τους χρήστες υπηρεσιών, πρέπει να δημιουργηθούν και οι Υπηρεσίες (Services) για την ολοκλήρωση της αυθεντικοποίησης.

Δημιουργία Υπηρεσιών

για την κάθε υπηρεσία χρησιμοποιείται η παρακάτω εντολή:

```
keystone service-create --name service_name --type service_type --description 'Description of the service'
```

```
keystone service-create --name ec2 --type ec2 --description 'EC2 Service'

keystone service-create --name nova --type compute --description 'OpenStack Compute Service'

keystone service-create --name keystone --type identity --description 'OpenStack Identity Service'

keystone service-create --name volume --type volume --description 'OpenStack Volume Service'

keystone service-create --name glance --type image --description 'OpenStack Image Service'

keystone service-create --name swift --type object-store --description 'OpenStack Storage Service'
```

Οι παρακάτω τιμές των πεδίων id τις κάθε λίστας πρόκειται να χρησιμοποιηθούν στο επόμενο βήμα, οπότε πρέπει να αποθηκευτούν σε ένα αρχείο κειμένου.

```
root@compute:~# keystone service-list
+-----+-----+-----+-----+
|          id          | name | type | description |
+-----+-----+-----+-----+
| 49a299a30a6d489ba64d0f0ef4ebfda6 | ec2  | ec2  | EC2 Service |
| 56b7d4ef9ced43908b32528a21b00b25 | nova  | compute | OpenStack Compute |
| 6a362837234f4f4db1d05afeae511e01 | keystone | identity | OpenStack Identity |
| 6a7c89954d1b409ab8a367794726493d | volume | volume | OpenStack Volume |
| c5da0162190f440782ae0194550b7284 | glance | image | OpenStack Image Service |
| e09523377add4ecf8e2548alc5838fb5 | swift  | object-store | OpenStack Storage |
+-----+-----+-----+-----+
```

Δημιουργία Endpoints

Κάθε υπηρεσία που έχει δημιουργηθεί, πρέπει να έχει ένα endpoint.

για το κάθε endpoint χρησιμοποιείται η παρακάτω εντολή:

```
keystone endpoint-create --region region_name --service_id service_id --
publicurl public_url --adminurl admin_url --internalurl internal_url
```

endpoint για το nova-compute:

```
keystone endpoint-create --region myregion --service_id
1e93ee6c70f8468c88a5cb1b106753f3 --publicurl
'http://10.10.10.2:8774/v2/$(tenant_id)s' --adminurl
'http://10.10.10.2:8774/v2/$(tenant_id)s' --internalurl
'http://10.10.10.2:8774/v2/$(tenant_id)s'
```

endpoint για το nova-volume:

```
keystone endpoint-create --region myregion --service_id
f38f4564ff7b4e43a52b2f5c1b75e5fa --publicurl
'http://10.10.10.2:8776/v1/$(tenant_id)s' --adminurl
'http://10.10.10.2:8776/v1/$(tenant_id)s' --internalurl
'http://10.10.10.2:8776/v1/$(tenant_id)s'
```

endpoint για το glance:

```
keystone endpoint-create --region myregion --service_id
fbafab6edcab467bb734380ce6be3561 --publicurl 'http://10.10.10.2:9292/v1' --
adminurl 'http://10.10.10.2:9292/v1' --internalurl
'http://10.10.10.2:9292/v1'
```

endpoint για το swift:

```
keystone endpoint-create --region myregion --service_id
96f35e1112b143e59d5cd5d0e6a8b22d --publicurl
'http://10.10.10.2:8080/v1/AUTH_$(tenant_id)s' --adminurl
'http://10.10.10.2:8080/v1' --internalurl
'http://10.10.10.2:8080/v1/AUTH_$(tenant_id)s'
```

endpoint για το keystone:

```
keystone endpoint-create --region myregion --service_id
7d4ec192dfa1456996f0f4c47415c7a7 --publicurl http://10.10.10.2:5000/v2.0 --
adminurl http://10.10.10.2:35357/v2.0 --internalurl
http://10.10.10.2:5000/v2.0
```

endpoint για το ec2:

```
keystone endpoint-create --region myregion --service_id
28fd92ffe3824004996a3e04e059d875 --publicurl
http://10.10.10.2:8773/services/Cloud --adminurl
http://10.10.10.2:8773/services/Admin --internalurl
http://10.10.10.2:8773/services/Cloud
```

Glance

Αρχικά θα παρουσιαστεί η βασική εγκατάσταση και παραμετροποίηση του Glance. Στο επόμενο κεφάλαιο θα γίνει η αναλυτική διαχείριση και μεταμόρφωση εικόνων.

εγκατάσταση του Glance

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

Για να επιτευχθεί η εξουσιοδότηση στον admin χρήστη glance του tenant service που δημιουργήθηκε παραπάνω, θα πρέπει να τροποποιηθούν τα αρχεία glance-api-paste.ini και glance-registry-paste.ini ανάλογα με τις τιμές που έχουν δοθεί.

Οι τιμές των μεταβλητών admin_tenant, admin_user και admin_password γίνονται :

```
admin_tenant_name = service
admin_user = glance
admin_password = glance
```

```
nano /etc/glance/glance-api-paste.ini
nano /etc/glance/glance-registry-paste.ini
```

Το glance έχει ως προεπιλογή διαχείρισης των βάσεων δεδομένων του την SQLite, στο αρχικό στάδιο επιλέχθηκε η MySQL. Θα χρησιμοποιηθούν οι τιμές username και password που δημιουργήθηκαν κατά την εγκατάσταση της MySQL.

Επεξεργασία του αρχείου glance-registry.conf για την χρήση MySQL

```
nano /etc/glance/glance-registry.conf
```

και η μεταβλητή sql_connection γίνεται:

```
sql_connection = mysql://glancedbadmin:glancesecret@10.10.10.2/glance
```

για να ρυθμιστεί η ταυτοποίηση του Glance μέσω του Keystone, επεξεργασία τις μεταβλητής flavor του αρχείου glance-api.conf.

```
nano /etc/glance/glance-api.conf
```

προσθήκη:

```
[paste_deploy]
flavor = keystone
```

Συγχρονισμός της βάσης του glance με MySQL

```
sudo glance-manage version_control 0
sudo glance-manage db_sync
```

Επανεκκίνηση του glance-api και glance-registry.

```
sudo restart glance-api
sudo restart glance-registry
```

Εξαγωγή των μεταβλητών εξουσιοδότησης ή προσθήκη στο αρχείο ~/.bashrc.

```
export SERVICE_TOKEN=admin
export OS_TENANT_NAME=admin
```



```
export OS_USERNAME=admin
export OS_PASSWORD=admin
export OS_AUTH_URL="http://localhost:5000/v2.0/"
export SERVICE_ENDPOINT=http://localhost:35357/v2.0
```

Έλεγχος για την ομαλή εγκατάσταση και ρύθμιση του Glance.

```
glance index
```

Αν δεν επιστρέψει καμία τιμή, τότε η εγκατάσταση είναι επιτυχής, και το Glance επικοινωνεί με το Keystone. Η ολοκλήρωση του Glance θα γίνει με την μεταμόρφωση εικόνων, που θα παρουσιαστεί αργότερα.

Nova

Εγκατάσταση του Nova

```
sudo apt-get install nova-api nova-cert nova-compute nova-compute-kvm nova-doc nova-network nova-objectstore nova-scheduler nova-volume rabbitmq-server novnc nova-consoleauth
```

Διαμόρφωση του Nova

Επεξεργασία του αρχείου `/etc/nova/nova.conf` σύμφωνα με το αρχείο που ακολουθεί.

```
--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--allow_admin_api=true
--use_deprecated_auth=false
--auth_strategy=keystone
--scheduler_driver=nova.scheduler.simple.SimpleScheduler
--s3_host=10.10.10.2
--ec2_host=10.10.10.2
--rabbit_host=10.10.10.2
--cc_host=10.10.10.2
--nova_url=http://10.10.10.2:8774/v1.1/
--routing_source_ip=10.10.10.2
--glance_api_servers=10.10.10.2:9292
--image_service=nova.image.glance.GlanceImageService
--iscsi_ip_prefix=192.168.4
--sql_connection=mysql://novadbadmin:novasecret@10.10.10.2/nova
--ec2_url=http://10.10.10.2:8773/services/Cloud
--keystone_ec2_url=http://10.10.10.2:5000/v2.0/ec2tokens
--api_paste_config=/etc/nova/api-paste.ini
--libvirt_type=kvm
--libvirt_use_virtio_for_bridges=true
--start_guests_on_host_boot=true
--resume_guests_state_on_host_boot=true
# vnc specific configuration
--novnc_enabled=true
--novncproxy_base_url=http://10.10.10.2:6080/vnc_auto.html
--vncserver_proxyclient_address=10.10.10.2
--vncserver_listen=0.0.0.0
# network specific settings
```

```
--network_manager=nova.network.manager.FlatDHCPManager
--public_interface=eth0
--flat_interface=eth1
--flat_network_bridge=br100
--fixed_range=192.168.4.1/27
--floating_range=10.10.10.2/27
--network_size=32
--flat_network_dhcp_start=192.168.4.33
--flat_injected=False
--force_dhcp_release
--iscsi_helper=tgtadm
--connection_type=libvirt
--root_helper=sudo nova-rootwrap
--verbose
```

Δημιουργούμε ένα Physical Volume.

```
sudo pvcreate /dev/sda6
```

Δημιουργία (volume group) ομάδας στο τόμο sda2 με όνομα nova-volumes.

```
sudo vgcreate nova-volumes /dev/sda2
```

Αλλαγή ιδιοκτήτη του φακέλου /etc/nova, προσθήκη του χρήστη nova.

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

Επεξεργασία δικαιωμάτων του αρχείου nova.conf, εγγραφή/ανάγνωση μόνο από τον ιδιοκτήτη και ανάγνωση από οποιονδήποτε.

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

Για να επιτευχθεί η εξουσιοδότηση από το Keystone στον admin χρήστη nova του tenant service θα πρέπει να τροποποιηθεί το αρχείο /etc/nova/api-paste.ini ανάλογα με τις τιμές που έχουν δοθεί.

Οι τιμές των μεταβλητών admin_tenant, admin_user και admin_password γίνονται :

```
admin_tenant_name = service
admin_user = nova
admin_password = nova
```

Δημιουργία σχήματος στην βάση δεδομένων της MySQL.

```
sudo nova-manage db sync
```

Παροχή εύρους IP για την σύνδεση των instances.

```
sudo nova-manage network create private --fixed_range_v4=192.168.4.32/27 --
num_networks=1 --bridge=br100 --bridge_interface=eth1 --network_size=32
```

Εξαγωγή των μεταβλητών εξουσιοδότησης ή προσθήκη στο αρχείο ~/.bashrc.

```
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin
export OS_AUTH_URL="http://localhost:5000/v2.0/"
```

Εκκίνηση του nova-volume, για την πρόληψη πιθανού σφάλματος 'unknown instances'.

```
sudo start nova-volume
```

Επανεκκίνηση όλων των υπηρεσιών του Nova.

```
sudo restart libvirt-bin; sudo restart nova-network; sudo restart nova-compute; sudo restart nova-api; sudo restart nova-objectstore; sudo restart nova-scheduler; sudo restart nova-volume; sudo restart nova-consoleauth;
```

Έλεγχος για την ομαλή εγκατάσταση και ρύθμιση του Nova.

```
sudo nova-manage service list
```

Binary	Host	Zone	Status	State	
Updated_At					
nova-network	server1	nova	enabled	:-)	2012-04-20 08:58:43
nova-scheduler	server1	nova	enabled	:-)	2012-04-20 08:58:44
nova-volume	server1	nova	enabled	:-)	2012-04-20 08:58:44
nova-compute	server1	nova	enabled	:-)	2012-04-20 08:58:45
nova-cert	server1	nova	enabled	:-)	2012-04-20 08:58:43

η στήλη State για όλες τις υπηρεσίες πρέπει να είναι :-)

Horizon

Εγκατάσταση του Dashboard

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

επανεκκίνηση του apache.

```
sudo service apache2 restart
```

Το Horizon έχει εγκατασταθεί και είναι έτοιμο για χρήση. Πηγαίνοντας στην διεύθυνση του server1 , 195.251.209.8 με έναν browser , εμφανίζεται η αρχική διεπαφή του Horizon για την σύνδεση χρήστη. Έχουμε χρησιμοποιήσει τα αλφαριθμητικά 'admin' για το όνομα χρήστη και 'admin' για τον κωδικό πρόσβασης. Παρακάτω θα γίνει η αναλυτική παρουσίαση της χρήσης του Horizon.

Swift

Εγκατάσταση του Swift.

```
sudo apt-get install swift swift-proxy swift-account swift-container swift-object
```

Εγκατάσταση επιπλέον απαραίτητων υπηρεσιών.

```
sudo apt-get install xfsprogs curl python-pastedeploy
```

Backend αποθήκευσης

Υπάρχουν δύο μέθοδοι για την δημιουργία του backend²⁴ αποθήκευσης του Swift. Ο πρώτος, είναι η χρήση ενός partition ή ολόκληρου τόμου ως μέσο αποθήκευσης και ο δεύτερος, είναι η δημιουργία loopback αρχείου²⁵. Εμείς χρησιμοποιούμε τον δεύτερο τρόπο.

Loopback file ως μέσο αποθήκευσης

Δημιουργούμε ένα κενό αρχείο για να το χρησιμοποιήσουμε ως backend για το Swift.

Δημιουργούμε ένα αρχείο με όνομα swift-disk με 1KiB blocks ο δίσκος είναι xfs.

```
sudo dd if=/dev/zero of=/srv/swift-disk bs=1024 count=0 seek=1000000  
sudo mkfs.xfs -i size=1024 /srv/swift-disk
```

Για να δούμε αν έγινε εκτελούμε:

```
file /srv/swift-disk
```

Χρησιμοποιώντας το Backend Αποθήκευσης

Προσάρτηση του swift backend.

```
sudo mount /mnt/swift_backend
```

Δημιουργία κόμβων που θα χρησιμοποιηθούν ως συσκευές αποθήκευσης.

```
pushd /mnt/swift_backend  
sudo mkdir node1 node2 node3 node4  
popd
```

Αλλαγή ιδιοκτησίας στον χρήστη και group 'swift'.

```
sudo chown swift.swift /mnt/swift_backend/*
```

Δημιουργία συνδέσμων για τους κόμβους.

```
for i in {1..4}; do sudo ln -s /mnt/swift_backend/node$i /srv/node$i; done;
```

Αλλαγή ιδιοκτησίας στον χρήστη και group 'swift', και δημιουργία καταλόγων για τους κόμβους.

```
sudo mkdir -p /etc/swift/account-server /etc/swift/container-server  
/etc/swift/object-server /srv/node1/device /srv/node2/device  
/srv/node3/device /srv/node4/device
```

²⁴ Με τον όρο backend, αναφέρεται το σύστημα υποστήριξης μιας υπηρεσίας, προγράμματος ή εφαρμογής που επικοινωνεί απευθείας με τους απαιτούμενους πόρους.

²⁵ Το Loopback αρχείο, αποτελεί ένα εικονικό σύστημα αρχείων που παρέχει εναλλακτική διαδρομή σε ένα υπάρχον σύστημα αρχείων. Δίνει την δυνατότητα να προσάπτονται σε αυτό διαφορετικά συστήματα αρχείων, χωρίς να τροποποιηθεί το αρχικό.

```
sudo mkdir /run/swift
sudo chown -L -R swift.swift /etc/swift /srv/node[1-4]/ /run/swift
```

Προσθήκη των παρακάτω γραμμών στο αρχείο `/etc/rc.local` , ακριβώς πριν το `'exit 0'` .

```
sudo mkdir /run/swift
sudo chown swift.swift /run/swift
```

Ρύθμιση του Rsync²⁶

Το Rsync είναι υπεύθυνο για τη διατήρηση αντιγράφων των αντικειμένων. Χρησιμοποιείται από τις υπηρεσίες του Swift για να διατηρείτε η συνοχή των αντικειμένων. Θα πρέπει να ρυθμιστεί για όλους τους κόμβους αποθήκευσης.

Ενεργοποίηση:

```
Set RSYNC_ENABLE=true in /etc/default/rsync.
```

Επεξεργασία του αρχείου `/etc/rsyncd.conf` , σύμφωνα με το παρακάτω:

```
# General stuff
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /run/rsyncd.pid
address = 127.0.0.1

# Account Server replication settings
[account6012]
max connections = 25
path = /srv/node1/
read only = false
lock file = /run/lock/account6012.lock

[account6022]
max connections = 25
path = /srv/node2/
read only = false
lock file = /run/lock/account6022.lock

[account6032]
max connections = 25
path = /srv/node3/
read only = false
lock file = /run/lock/account6032.lock

[account6042]
max connections = 25
path = /srv/node4/
read only = false
lock file = /run/lock/account6042.lock

# Container server replication settings

[container6011]
max connections = 25
```

²⁶ Rsync είναι ένα πρωτόκολλο δικτύωσης που συγχρονίζει αρχεία και καταλόγους.

```
path = /srv/node1/
read only = false
lock file = /run/lock/container6011.lock

[container6021]
max connections = 25
path = /srv/node2/
read only = false
lock file = /run/lock/container6021.lock

[container6031]
max connections = 25
path = /srv/node3/
read only = false
lock file = /run/lock/container6031.lock

[container6041]
max connections = 25
path = /srv/node4/
read only = false
lock file = /run/lock/container6041.lock

# Object Server replication settings

[object6010]
max connections = 25
path = /srv/node1/
read only = false
lock file = /run/lock/object6010.lock

[object6020]
max connections = 25
path = /srv/node2/
read only = false
lock file = /run/lock/object6020.lock

[object6030]
max connections = 25
path = /srv/node3/
read only = false
lock file = /run/lock/object6030.lock

[object6040]
max connections = 25
path = /srv/node4/
read only = false
lock file = /run/lock/object6040.lock
```

επανεκκίνηση rsync.

```
sudo service rsync restart
```

Ρύθμιση των υπηρεσιών του Swift

Το Python χρησιμοποιεί το σύστημα `paste.deploy` για την διαχείριση των WSGH²⁷ διακομιστών και εφαρμογών. Οι προεπιλεγμένες ρυθμίσεις βρίσκονται ανάμεσα σε αγκύλες [], ενώ για οποιαδήποτε αλλαγή σε κάποια επιλογή γίνεται η χρήση της σύνταξης `set option_name = value`.

Για παράδειγμα:

```
# [DEFAULT]
# name1 = globalvalue
# set name2 = globalvalue

[pipeline:main]
pipeline = myapp

[app:myapp]
use = egg:mypkg#myapp
name2 = localvalue
set name3 = localvalue
```

Επεξεργασία του αρχείου `/etc/swift/swift.conf` και προσθήκη:

```
[swift-hash]
# random unique string that can never change (DO NOT LOSE). I'm using
03c9f48da2229770.
# od -t x8 -N 8 -A n < /dev/random
# The above command can be used to generate random a string.
swift_hash_path_suffix = 03c9f48da2229770
```

Κατά την πρόσθεση επιπλέον κόμβων στην εγκατάσταση θα χρειαστεί το τυχαίο string που εκχωρήθηκε στην μεταβλητή `swift_hash_path_suffix`. Γενικά μπορούν να παραχθούν τυχαία strings 16 ψηφίων με την εντολή:

```
od -t x8 -N 8 -A n < /dev/random
```

Ρύθμιση του Swift Proxy Server

Επεξεργασία του αρχείου `/etc/swift/proxy-server.conf` και προσθήκη:

```
[DEFAULT]
bind_port = 8080
user = swift
swift_dir = /etc/swift

[pipeline:main]
# Order of execution of modules defined below
pipeline = catch_errors healthcheck cache authtoken keystone proxy-server

[app:proxy-server]
use = egg:swift#proxy
allow_account_management = true
```

²⁷ WSGH(Web Server Gateway Interface) ορίζει μια απλή διεπαφή ανάμεσα σε Web διακομιστές και Web εφαρμογές ή πλαίσια για την Python.

```

account_autocreate = true
set log_name = swift-proxy
set log_facility = LOG_LOCAL0
set log_level = INFO
set access_log_name = swift-proxy
set access_log_facility = SYSLOG
set access_log_level = INFO
set log_headers = True
account_autocreate = True

[filter:healthcheck]
use = egg:swift#healthcheck

[filter:catch_errors]
use = egg:swift#catch_errors

[filter:cache]
use = egg:swift#memcache
set log_name = cache

[filter:authtoken]
paste.filter_factory = keystone.middleware.auth_token:filter_factory
auth_protocol = http
auth_host = 127.0.0.1
auth_port = 35357
auth_token = admin
service_protocol = http
service_host = 127.0.0.1
service_port = 5000
admin_token = admin
admin_tenant_name = service
admin_user = swift
admin_password = swift
delay_auth_decision = 0

[filter:keystone]
paste.filter_factory = keystone.middleware.swift_auth:filter_factory
operator_roles = admin, swiftoperator
is_admin = true

```

Ρύθμιση του Swift Account Server

Επεξεργασία του αρχείου `/etc/swift/account-server.conf`. Για να δημιουργηθεί σύνδεση με τις υπηρεσίες-servers του Swift, η `bind_ip` πρέπει να έχει την `ip` του server που είναι εγκαταστημένο το Swift.

```

[DEFAULT]
bind_ip = 0.0.0.0
workers = 2

[pipeline:main]
pipeline = account-server

[app:account-server]
use = egg:swift#account

```



```
[account-replicator]

[account-auditor]

[account-reaper]
```

Για κάθε κόμβο αποθήκευσης που έχει υπάρξει στην διαδρομή /srv/node(1..4) ,πρέπει να δημιουργηθεί ένα αρχείο παρόμοιο με το account-server.conf. Άρα δημιουργούνται τέσσερα αρχεία, 1.conf,2.conf,3.conf, και 4.conf ,στην διαδρομή /etc/swift/account-server/. Παρακάτω παρουσιάζονται τα περιεχόμενα των αρχείων.

1.conf

```
[DEFAULT]
devices = /srv/node1
mount_check = false
bind_port = 6012
user = swift
log_facility = LOG_LOCAL2

[pipeline:main]
pipeline = account-server

[app:account-server]
use = egg:swift#account

[account-replicator]
vm_test_mode = no

[account-auditor]

[account-reaper]
```

2.conf

```
DEFAULT]
devices = /srv/node2
mount_check = false
bind_port = 6022
user = swift
log_facility = LOG_LOCAL3
[pipeline:main]
pipeline = account-server
[app:account-server]
use = egg:swift#account
[account-replicator]
vm_test_mode = no
[account-auditor]
[account-reaper]
```

3.conf

```
DEFAULT]
```

```

devices = /srv/node3
mount_check = false
bind_port = 6032
user = swift
log_facility = LOG_LOCAL4
[pipeline:main]
pipeline = account-server
[app:account-server]
use = egg:swift#account
[account-replicator]
vm_test_mode = no
[account-auditor]
[account-reaper]

```

4.conf

```

[DEFAULT]
devices = /srv/node4
mount_check = false
bind_port = 6042
user = swift
log_facility = LOG_LOCAL5
[pipeline:main]
pipeline = account-server
[app:account-server]
use = egg:swift#account
[account-replicator]
vm_test_mode = no
[account-auditor]
[account-reaper]

```

Ρύθμιση του Swift Container Server

Επεξεργασία του αρχείου `/etc/swift/container-server.conf` . Υπενθύμιση, ότι η `bind_ip` πρέπει να έχει την `ip` του `server1` .

`/etc/swift/container-server.conf:`

```

[DEFAULT]
bind_ip = 0.0.0.0
workers = 2

[pipeline:main]
pipeline = container-server

[app:container-server]
use = egg:swift#container

[container-replicator]

[container-updater]

[container-auditor]

[container-sync]

```

Για κάθε κόμβο, πρέπει να δημιουργηθεί ένα αρχείο παρόμοιο με το container-server.conf. Άρα δημιουργούνται τέσσερα αρχεία, 1.conf, 2.conf, 3.conf, και 4.conf, στην διαδρομή /etc/swift/container-server/. Παρακάτω παρουσιάζονται τα περιεχόμενα των αρχείων.

1.conf

```
[DEFAULT]
devices = /srv/node1
mount_check = false
bind_port = 6011
user = swift
log_facility = LOG_LOCAL2
[pipeline:main]
pipeline = container-server
[app:container-server]
use = egg:swift#container
[container-replicator]
vm_test_mode = no
[container-updater]
[container-auditor]
[container-sync]
```

2.conf

```
[DEFAULT]
devices = /srv/node2
mount_check = false
bind_port = 6021
user = swift
log_facility = LOG_LOCAL3
[pipeline:main]
pipeline = container-server
[app:container-server]
use = egg:swift#container
[container-replicator]
vm_test_mode = no
[container-updater]
[container-auditor]
[container-sync]
```

3.conf

```
[DEFAULT]
devices = /srv/node3
mount_check = false
bind_port = 6031
user = swift
log_facility = LOG_LOCAL4
[pipeline:main]
pipeline = container-server
[app:container-server]
use = egg:swift#container
[container-replicator]
vm_test_mode = no
[container-updater]
[container-auditor]
[container-sync]
```

4.conf

```
[DEFAULT]
devices = /srv/node4
mount_check = false
bind_port = 6041
user = swift
log_facility = LOG_LOCAL5
[pipeline:main]
pipeline = container-server
[app:container-server]
use = egg:swift#container
[container-replicator]
vm_test_mode = no
[container-updater]
[container-auditor]
[container-sync]
```

Ρύθμιση του Swift Object Server

Για κάθε κόμβο, πρέπει να δημιουργηθεί ένα αρχείο παρόμοιο με το object-server.conf. Άρα δημιουργούνται τέσσερα αρχεία, 1.conf, 2.conf, 3.conf, και 4.conf, στην διαδρομή /etc/swift/container-server/. Παρακάτω παρουσιάζονται τα περιεχόμενα των αρχείων.

/etc/swift/object-server.conf :

```
[DEFAULT]
bind_ip = 0.0.0.0
workers = 2

[pipeline:main]
pipeline = object-server

[app:object-server]
use = egg:swift#object

[object-replicator]

[object-updater]

[object-auditor]
```

/etc/swift/object-server/1.conf :

```
[DEFAULT]
devices = /srv/node1
mount_check = false
bind_port = 6010
user = swift
log_facility = LOG_LOCAL2
[pipeline:main]
pipeline = object-server
[app:object-server]
use = egg:swift#object
[object-replicator]
```

```
vm_test_mode = no
[object-updater]
[object-auditor]
```

/etc/swift/object-server/2.conf :

```
[DEFAULT]
devices = /srv/node2
mount_check = false
bind_port = 6020
user = swift
log_facility = LOG_LOCAL3
[pipeline:main]
pipeline = object-server
[app:object-server]
use = egg:swift#object
[object-replicator]
vm_test_mode = no
[object-updater]
[object-auditor]
```

/etc/swift/object-server/3.conf :

```
[DEFAULT]
devices = /srv/node3
mount_check = false
bind_port = 6030
user = swift
log_facility = LOG_LOCAL4
[pipeline:main]
pipeline = object-server
[app:object-server]
use = egg:swift#object
[object-replicator]
vm_test_mode = no
[object-updater]
[object-auditor]
```

/etc/swift/object-server/4.conf :

```
[DEFAULT]
devices = /srv/node4
mount_check = false
bind_port = 6040
user = swift
log_facility = LOG_LOCAL5
[pipeline:main]
pipeline = object-server
[app:object-server]
use = egg:swift#object
[object-replicator]
vm_test_mode = no
[object-updater]
[object-auditor]
```

Ρύθμιση των Swift Rings

Δημιουργία των αρχείων ring builder για τις υπηρεσίες object, container και account με την εντολή :

```
swift-ring-builder <builder_file> create <part_power> <replicas>
<min_part_hours>.
```

Οι παράμετροι part_power , replicas και min_part_hours, αντιπροσωπεύουν τον αριθμό των διαμερισμάτων, των αντιγράφων και των ωρών που απαγορεύεται η μετακίνηση ενός διαμερίσματος πάνω απο μια φορά.

εκτέλεση των εντολών γίνεται στην διαδρομή /etc/swift

```
pushd /etc/swift
sudo swift-ring-builder object.builder create 18 3 1
sudo swift-ring-builder container.builder create 18 3 1
sudo swift-ring-builder account.builder create 18 3 1
```

Πρόσθεση συσκευών των κόμβων που δημιουργήθηκαν , σε κάθε αρχείο ring :

```
swift-ring-builder <builder_file> add <zone>-<ip_address>:<port>/<device>
<weight>
```

Το zone : αντιπροσωπεύει τον κόμβο της συσκευής , port : το bind_port του κόμβου , device : όνομα συσκευής και weight : αριθμός συσκευών. Με το rebalance γίνεται η αντιστοίχιση των διαμερισμάτων στην συσκευή που έχει προστεθεί .

```
sudo swift-ring-builder object.builder add z1-127.0.0.1:6010/device 1 sudo
swift-ring-builder object.builder add z2-127.0.0.1:6020/device 1 sudo swift-
ring-builder object.builder add z3-127.0.0.1:6030/device 1 sudo swift-ring-
builder object.builder add z4-127.0.0.1:6040/device 1 sudo swift-ring-builder
object.builder rebalance
sudo swift-ring-builder container.builder add z1-127.0.0.1:6011/device 1
sudo swift-ring-builder container.builder add z2-127.0.0.1:6021/device 1
sudo swift-ring-builder container.builder add z3-127.0.0.1:6031/device 1
sudo swift-ring-builder container.builder add z4-127.0.0.1:6041/device 1
sudo swift-ring-builder container.builder rebalance
sudo swift-ring-builder account.builder add z1-127.0.0.1:6012/device 1 sudo
swift-ring-builder account.builder add z2-127.0.0.1:6022/device 1 sudo swift-
ring-builder account.builder add z3-127.0.0.1:6032/device 1 sudo swift-ring-
builder account.builder add z4-127.0.0.1:6042/device 1 sudo swift-ring-
builder account.builder rebalance
```

Εκκίνηση των υπηρεσιών Swift

```
sudo swift-init main start
sudo swift-init rest start
```

Αλλαγή ιδιοκτησίας της διαδρομής /etc/swift στον χρήστη και group swift :

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

Για να ελεγχθεί αν το swift λειτουργεί σωστά, χρησιμοποιήθηκε το dashboard. Η είσοδος πρέπει να γίνει με τον χρήστη swift. Στην καρτέλα Project στο κάτω μέρος υπάρχει το τμήμα Object Store, εκεί γίνεται η διαχείριση των containers μαζί με τα object τους. Δημιουργήθηκε ένα container που έχει ένα object. Και σύμφωνα με την παρακάτω έντολη φαίνεται ότι η εγκατάσταση του swift έγινε επιτυχώς

```
swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U service:swift -K swift stat
StorageURL: http://195.251.209.8:8080/v1/AUTH_99c9648fed41445e86e4452d75b6cb5a
Auth Token: a2ac0f49a3be423c9f5964f672bafbb9
Account: AUTH_99c9648fed41445e86e4452d75b6cb5a
Containers: 1
Objects: 1
Bytes: 704574
Accept-Ranges: bytes
X-Trans-Id: txff9f55c8900a4e75b54451aba05217ee
```

2. Server2

Εγκατάσταση

Ο Server2 , περιέχει μόνο την υπηρεσία nova-compute του OpenStack. Το λειτουργικό σύστημα που τρέχει είναι η 64 bit έκδοση της διανομής Ubuntu server 12.04. Τα βασικά σημεία κατά την εγκατάσταση είναι η χειροκίνητη ρύθμιση των IP σύμφωνα με τις τιμές του Πίνακα εγκατάστασης και η επιλογή μόνο του πακέτου Openssh-server στο Packages menu.

Βασικές ενέργειες

Ενημέρωση του συστήματος

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install bridge-utils
```

Ρυθμίσεις Δικτύου

Επεξεργασία του αρχείου interfaces στην παρακάτω μορφή

```
nano /etc/network/interfaces
```

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 195.251.209.7
```

```

netmask 255.255.255.192
network 195.251.209.0
broadcast 195.251.209.63
gateway 195.251.209.1
# dns-* options are implemented by the resolvconf package, if installed
dns-nameservers 195.251.213.106 195.251.213.104
dns-search it.uom.gr

iface eth1 inet static
    address 192.168.0.2
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255

```

Επανεκκίνηση δικτύου

```
sudo /etc/init.d/networking restart
```

Πελάτης NTP

Εγκατάσταση πακέτου NTP

```
sudo apt-get install ntp
```

προσθήκη στο αρχείο /etc/ntp.conf για τον συγχρονισμό με τον server1

```
server 195.251.209.8
```

επανεκκίνηση της υπηρεσίας NTP.

```
sudo service ntp restart
```

Nova

Εγκατάσταση του nova-compute

```
sudo apt-get install nova-compute
```

επεξεργασία του αρχείου /etc/nova/nova.conf, το οποίο θα πρέπει να είναι το ίδιο με το nova.conf του server1

```

--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--allow_admin_api=true
--use_deprecated_auth=false
--auth_strategy=keystone
--scheduler_driver=nova.scheduler.simple.SimpleScheduler
--s3_host=195.251.209.8
--ec2_host=195.251.209.8
--rabbit_host=195.251.209.8
--cc_host=195.251.209.8
--nova_url=http://195.251.209.8:8774/v1.1/

```



```

--routing_source_ip=195.251.209.8
--glance_api_servers=195.251.209.8:9292
--image_service=nova.image.glance.GlanceImageService
--iscsi_ip_prefix=192.168.1
--sql_connection=mysql://novadbadmin:novasecret@195.251.209.8/nova
--ec2_url=http://195.251.209.8:8773/services/Cloud
--keystone_ec2_url=http://195.251.209.8:5000/v2.0/ec2tokens
--api_paste_config=/etc/nova/api-paste.ini
--libvirt_type=kvm
--libvirt_use_virtio_for_bridges=true
--start_guests_on_host_boot=true
--resume_guests_state_on_host_boot=true
# vnc specific configuration
--novnc_enabled=true
--novncproxy_base_url=http://195.251.209.8:6080/vnc_auto.html
--vncserver_proxyclient_address=195.251.209.8
--vncserver_listen=0.0.0.0
# network specific settings
--network_manager=nova.network.manager.FlatDHCPManager
--public_interface=eth0
--flat_interface=eth1
--flat_network_bridge=br100
--fixed_range=192.168.1.1/27
--floating_range=195.251.209.8/27
--network_size=32
--flat_network_dhcp_start=192.168.1.33
--flat_injected=False
--force_dhcp_release
--iscsi_helper=tgtadm
--connection_type=libvirt
--root_helper=sudo nova-rootwrap
--verbose

```

επανεκκίνηση του nova-compute

```
sudo service nova-compute restart
```

έλεγχος αν η ανίχνευση του server2 στην εγκατάσταση έγινε με επιτυχία

```
sudo nova-manage service list
```

Binary	Host	Zone	Status	State
Updated At				
nova-scheduler	compute	nova	enabled	: -)
2012-09-02 21:38:56				
nova-compute	compute	nova	enabled	: -)
2012-09-02 21:38:56				
nova-consoleauth	compute	nova	enabled	: -)
2012-09-02 21:38:48				
nova-network	compute	nova	enabled	: -)
2012-09-02 21:38:49				
nova-volume	compute	nova	enabled	: -)
2012-09-02 21:38:49				
nova-cert	compute	nova	enabled	: -)
2012-09-02 21:38:49				
nova-compute	Controller	nova	enabled	: -)
2012-09-02 21:38:48				

3. Client1

Εγκατάσταση

Ο Client1, δεν περιέχει καμία υπηρεσία του OpenStack, παρά μόνο κάποια εργαλεία διαχείρισης του nova, glance και swift. Το λειτουργικό σύστημα που τρέχει είναι η 64 bit έκδοση της διανομής Ubuntu desktop 12.04.

Βασικές ενέργειες

Ενημέρωση του συστήματος

```
sudo apt-get update  
sudo apt-get upgrade
```

Ρυθμίσεις Δικτύου

Επεξεργασία του αρχείου interfaces στην παρακάτω μορφή

```
nano /etc/network/interfaces
```

```
auto lo  
    iface lo inet loopback  
auto eth0  
    iface eth0 inet static  
    address 195.251.209.6  
    netmask 255.255.255.192  
    network 195.251.209.0  
    broadcast 195.251.209.63  
    gateway 195.251.209.1  
    dns-nameservers 195.251.213.106 195.251.213.104
```

Επανεκκίνηση δικτύου

```
sudo /etc/init.d/networking restart
```

Πελάτης NTP

Εγκατάσταση πακέτου NTP

```
sudo apt-get install -y ntp
```

προσθήκη στο αρχείο /etc/ntp.conf για τον συγχρονισμό με τον server1

```
server 195.251.209.8
```

επανεκκίνηση της υπηρεσίας NTP.

```
sudo service ntp restart
```

Εργαλεία Πελάτη

Η κύρια χρήση του πελάτη στην παρούσα εγκατάσταση, είναι το bundling εικόνων και η διαχείριση της υποδομής του «νέφους» με ορισμένα εργαλεία τερματικού που παρέχει το swift, glance και nova.

Εγκατάσταση εργαλείων τερματικού :

```
sudo apt-get install python-novaclient glance-client swift
```

Εγκατάσταση KVM²⁸

```
sudo apt-get install qemu-kvm
```

Εξαγωγή των μεταβλητών εξουσιοδότησης ή προσθήκη στο αρχείο ~/.bashrc.

```
export SERVICE_TOKEN=admin
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin
export OS_AUTH_URL="http://195.251.209.8/:5000/v2.0/"
export SERVICE_ENDPOINT=http://195.251.209.8/:35357/v2.0
```

Έλεγχος για την επιτυχή σύνδεση του πελάτη στην εγκατάσταση δια μέσο του nova και glance.

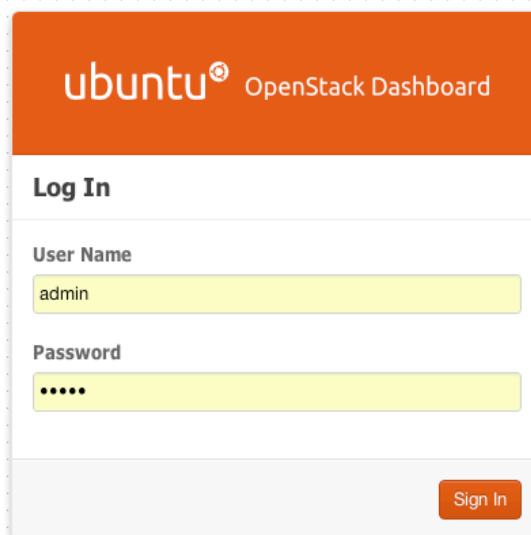
```
nova list
+-----+-----+-----+-----+ | ID | Name | Status | Networks |
+-----+-----+-----+-----+
+-----+-----+-----+-----+

glance index
```

Αφού δεν έχουν δημιουργηθεί ακόμη instances και εικόνες ,ο πίνακας nova list θα είναι φυσικά κενός και το glance index δεν πρέπει να επιστρέψει καμία τιμή.

OpenStack Dashboard

Ανοίγοντας έναν browser με διεύθυνση την ip του server1 (195.251.209.8), εμφανίζεται η είσοδος χρήστη στην γραφική διεπαφή διαχείρισης του «νέφους».



ubuntu[®] OpenStack Dashboard

Log In

User Name
admin

Password
.....

Sign In

5. Διαχείριση Εικόνων

Εισαγωγή

Η εικόνα στον χώρο της νεφοπληροφορικής αναπαριστά έναν δίσκο χωρίς partition στον οποίο υπάρχει το λειτουργικό σύστημα που τρέχει μια εικονική μηχανή. Υπάρχουν έτοιμες προ-εγκατεστημένες εικόνες που χρησιμοποιούνται κυρίως για τον έλεγχο του συστήματος (pre-built) , αλλά και εικόνες που παρέχονται μετά από bundling όπως προαναφέρθηκε στην αρχή του τέταρτου κεφαλαίου, δηλαδή την ολοκληρωμένη εγκατάσταση και διαμόρφωση ενός λειτουργικού συστήματος σε ένα εικονικό αρχείο.

Οι διανομές του Linux όπως Ubuntu, Fedora και OpenSUSE κυριαρχούν στον χώρο. Ειδικότερα το Ubuntu με το πακέτο cloud-init, παρέχει μια φιλικότερη διαχείριση των instances όπως θα παρουσιαστεί παρακάτω. Τέλος, και οι διανομές της Microsoft συμβάλουν στην υπηρεσία εικόνων , χρησιμοποιώντας οδηγούς εγκατάστασης Virtio το οποίο παρέχει πλήρης εικονικοποίηση συστήματος. Το μειονέκτημα από την χρήση του Virtio είναι η μείωση ταχύτητας , γεγονός που καθιστά τις διανομές linux προτιμότερες.

Εγκατάσταση Λειτουργικού Συστήματος

Θα παρουσιαστεί ως παράδειγμα, η εγκατάσταση του Ubuntu Server 12.04 (<http://releases.ubuntu.com>) . Αρχικά δημιουργείτε το αρχείο εικόνας που αναπαριστά τον σκληρό δίσκο της εικονικής μηχανής.

```
kvm-img create -f qcow2 server.img 5G
```

Με την χρήση του kvm γίνεται η εκκίνηση της εγκατάστασης του αρχείου iso από ένα εικονικό cd-rom και ξεκινά η εγκατάσταση του OS.

```
sudo kvm -m 256 -cdrom ubuntu-12.04-server-amd64.iso -drive file=server.img,if=virtio,index=0 -boot d -net nic -net user -nographic -vnc :0
```

Η εικονική μηχανή συνδέεται με το VNC στην ip του client1 στην πόρτα 0 για να ολοκληρωθεί η εγκατάσταση.

```
vncviewer 195.251.209.6:0
```

Κατά την εγκατάσταση πρέπει να δημιουργηθεί ένα partition ext4 για το swapping (ιδανικό μέγεθος 1-2GB). Αφού τελειώσει η εγκατάσταση η εικονική μηχανή είναι έτοιμη προς χρήση με εγκατεστημένο λειτουργικό σύστημα.

```
sudo kvm -m 256 -drive file=server.img,if=virtio,index=0 -boot c -net nic -  
net user -nographic -vnc :0
```

απαραίτητες ενέργειες:

```
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install openssh-server cloud-init  
sudo rm -rf /etc/udev/rules.d/70-persistent-net.rules
```

Μεταμόρφωση εικόνων

Το ανέβασμα των εικόνων γίνεται δια μέσου του Client1 , με την χρήση του πακέτου glance-client. Η εικόνα αποθηκεύεται στον server όπου είναι εγκατεστημένο το glance, άρα στον server1.

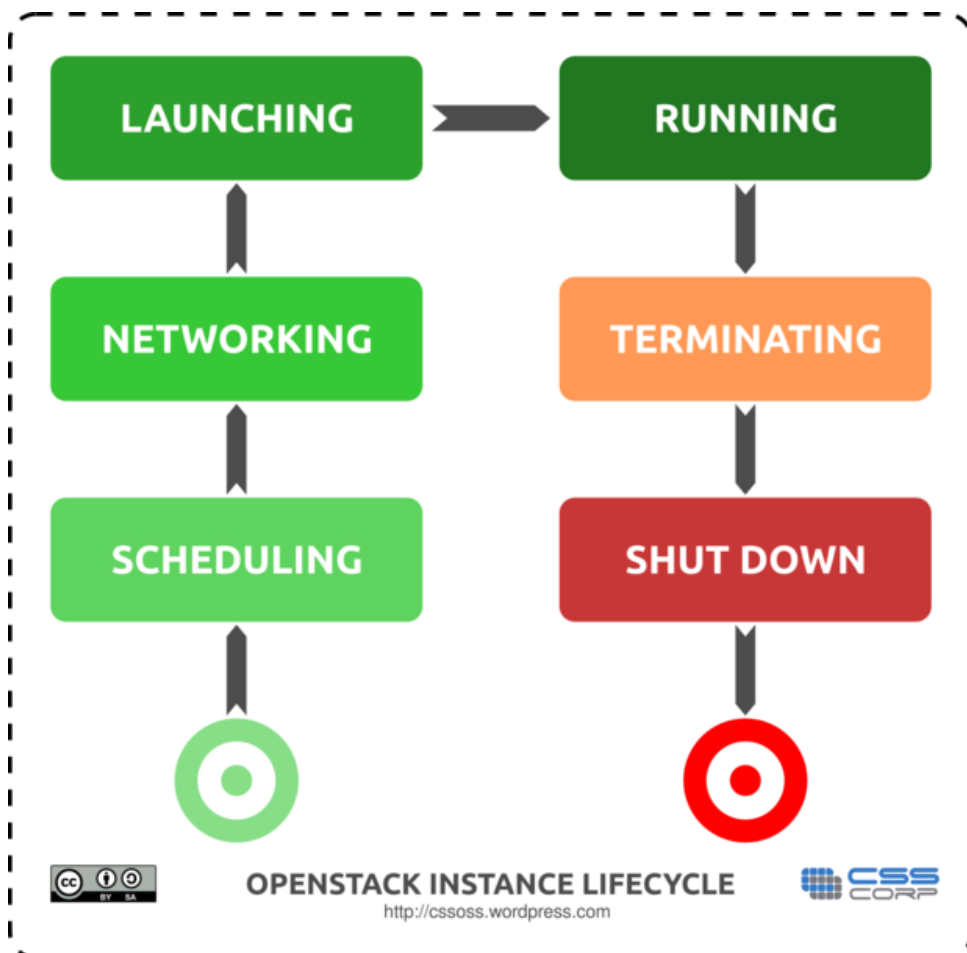
Ανέβασμα εικόνας:

```
glance add name="<Image name>" is_public=true container_format=ovf  
disk_format=qcow2 < <filename>.img
```

6. Διαχείριση των Instances

Εισαγωγή

Ένα instance αποτελεί μια εικονική μηχανή που παρέχει η υπηρεσία nova-compute του server2. Οι καταστάσεις στις οποίες βρίσκεται ένα instance ,είναι γνωστές ως κύκλος ζωής των instances όπως φαίνεται στην παρακάτω εικόνα.



Ρύθμιση των floating IP's

Αμέσως μετά την δημιουργία του instance ,παρέχεται μια private ip. Η ip όμως με την οποία επικοινωνεί με τον έξω κόσμο(public) ονομάζεται floating ip και του προσάπτεται δυναμικά . Αρχικά δημιουργείται ένα εύρος διευθύνσεων με την μέθοδο δρομολόγησης CIDR ή απευθείας απόδοση διευθύνσεων ip , και στην συνέχεια κάθε instance αποκτά και μία μοναδική δημόσια ip. Σημειώνουμε ότι η λίστα με τις διαθέσιμες ip ονομάζεται pool. Κάθε pool δημιουργείται από τον διαχειριστή μόνο μέσω της κονσόλας του server1 , και είναι διαθέσιμο προς όλους τους χρήστες στο Dashboard, στο οποίο υπάρχει η δυνατότητα στον χρήστη να διαθέσει ή να αλλάξει την floating ip ενός instance. Τέλος, η floating ip ορίζεται με τα network bits (24 πρώτα bits) τις ip του server 2 , και τα υπόλοιπα host bits να ανήκουν σε κάθε υπολογιστή υπηρεσίας (instance).

Δημιουργία δημοσίων διευθύνσεων , σύμφωνα με τον ορισμό που δόθηκε παραπάνω έχουμε:

```
nova-manage floating create --ip_range=195.251.209.14 --pool=nova
nova-manage floating create --ip_range=195.251.209.15 --pool=nova
nova-manage floating create --ip_range=195.251.209.16 --pool=nova
```

Δημιουργία pool με όνομα nova, για την προσάρτηση κάποιας από της διαθέσιμες ip (οπου novapublicip αποτελεί το όνομα του pool)

```
nova floating-ip-create nova
```

Προσοχή!

Σε αυτό το σημείο πρέπει να δημιουργηθεί ένα instance , με τον τρόπο που παρουσιάζεται παρακάτω. Έστω οτι έχουμε ένα έτοιμο.

Πρόσθεση floating ip στο τρέχων instance.

```
nova add-floating-ip 23efa3ec-ed5e-4c59-b4ed-fe5dddc60684 195.251.209.14
nova floating-ip-list
```

Ip	Instance Id	Fixed Ip	Pool
195.251.209.14	23efa3ec-ed5e-4c59-b4ed-fe5dddc60684	192.168.0.34	nova
195.251.209.15	None	None	nova
195.251.209.16	None	None	nova

Σημειώνεται ότι: μετά την πρώτη διευθυνσιοδότηση μιας floating ip από ένα pool σε ένα instance, οι επόμενες γίνονται αυτόματα στις αμέσως επόμενες ip για κάθε νέο instance που δημιουργείτε.

Δημιουργία των Key pairs

Για να αποκτήσει πρόσβαση ο διαχειριστής στο instance , θα πρέπει αμέσως μετά το πρώτο boot να συνδεθεί με SSH χωρίς την χρήση password. Η κίνηση αυτή επιτυγχάνεται με την χρήση των key pairs. Κάθε key pair αποτελείται από ένα ιδιωτικό και ένα δημόσιο κλειδί, στο οποίο κατά την δημιουργία του ο διαχειριστής χρησιμοποιεί ένα δικό του κωδικό(phrase), τον οποίο τον χρησιμοποιεί και έπειτα κατά την σύνδεση με SSH.

Δημιουργία

```
ssh-keygen
cd .ssh
nova keypair-add --pub_key id_rsa.pub mykey
chmod 600 mykey.pem
nova keypair-list
```

Name	Fingerprint
mykey	aa:43:c8:aa:e7:03:44:17:f6:33:9d:cb:e9:30:f5:26

Δημιουργία των Security Groups

Τα Security Groups είναι ομάδες κανόνων και χρησιμοποιούνται για να προσθήκη εξαιρέσεων όπως ένα firewall. Εμείς θα προσθέσουμε όλες τις εξαιρέσεις για να έχει το instance πρόσβαση σε όλο το internet.

```
nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
```

Εκκίνηση των Instances

Ένα instance μπορεί να διαχειριστεί είτε από το τερματικό είτε από το dashboard. Χρησιμοποιώντας το τερματικό για την εκκίνηση ενός instance έχουμε

```
nova boot --flavor < επιλογή τύπου flavor> --image <uuid του image> --key_name <όνομα key pair> <όνομα instance>
```

```
nova boot --flavor 1 --image 45bd2d24-2079-4c25-937f-3b08b11061c6 --key_name mykey
myinstance
```

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-SRV-ATTR:host	Controller
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	instance-00000006
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building

accessIPv4	
accessIPv6	
adminPass	6KY5mMp3EekP
config_drive	
created	2012-10-27T18:39:07Z
flavor	m1.tiny
hostId	
id	23efa3ec-ed5e-4c59-b4ed-fe5dddc60684
image	Ubuntu Server test
key_name	mykey
metadata	{}
name	myinstance
progress	0
status	BUILD
tenant_id	a712d9bd1fed4b3ba607652e67cea441
updated	2012-10-27T18:39:07Z
user_id	c3e95bdbc5b84e648205eea540ec7e94

```
nova list
+-----+-----+-----+-----+
+          ID              | Name      | Status | Networks
+-----+-----+-----+-----+
+ 23efa3ec-ed5e-4c59-b4ed-fe5dddc60684 | myinstance | ACTIVE | private=192.168.0.34,
195.251.209.14 |
+-----+-----+-----+-----+
```

Μετά την εκκίνηση του instance είτε γίνει από την κονσόλα είτε από το dashboard, θα πρέπει να γίνει SSH στην private ip του instance, για την απόκτηση πρόσβασης. Υπενθυμίζεται ότι η παρακάτω μέθοδος (password-less ssh) γίνεται για τον πλήρη έλεγχο του instance, χωρίς την χρήση credentials.

```
root@Controller:~# cd ~/.ssh/
root@Controller:~/.ssh# ssh -i id_rsa ubuntu@192.168.0.34
Enter passphrase for key 'id_rsa':

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Welcome to Ubuntu quantal (development branch) (GNU/Linux 3.5.0-13-generic
x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Sat Oct 27 18:54:41 UTC 2012

System load:  0.0          Processes:           64
Usage of /:   31.2% of 1.96GB Users logged in:      0
```

```

Memory usage: 8%                               IP address for eth0: 192.168.0.34
Swap usage:   0%

Graph this data and manage this system at https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

Get cloud support with Ubuntu Advantage Cloud Guest
http://www.ubuntu.com/business/services/cloud
ubuntu@myinstance:~$

```

και στην συνέχεια προσθέτουμε χρήστες , για παράδειγμα

```

ubuntu@myinstance:~$ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully

```

Δικαιώματα χρηστών

Σύμφωνα με την ενότητα του Keystone , η διαχείριση των χρηστών γίνεται βάση του tenant στο οποίο ανήκουν και τον ρόλο που έχουν σε αυτό. Κάθε υπηρεσία προσφέρει την δυνατότητα διαχείρισης των δικαιωμάτων που έχει το κάθε tenant. Η οριοθέτηση πρόσβασης γίνεται στο αρχείο policy.json που βρίσκεται στο κατάλογο etc/φακελος-υπηρεσίας. Κάθε αρχείο policy έχει ως προεπιλογή ανοικτά όλα τα δικαιώματα για κάθε χρήστη. Η παραμετροποίηση γίνεται με την εντολή και δίπλα τον ρόλο ενός χρήστη που θέλουμε να του δώσουμε πρόσβαση. Για παράδειγμα αν θέλουμε να δώσουμε το δικαίωμα δημιουργίας και προσάρτησης volumes, σε έναν χρήστη που έχει ρόλο member σε ένα tenant τότε έχουμε : επειδή η επιλογές των volumes γίνονται από το nova άρα θα επεξεργαστούμε το αρχείο /etc/nova/policy.json προσθέτοντας

```

"compute:create": [ "role": "member" ],
"compute:create:attach_volume": [ "role": "member" ],

```

Προσαρμογή του Dashboard

Η διεπαφή διαχείρισης που προσφέρει η OpenStack μπορεί να τροποποιηθεί και να προσαρμοστεί στις απαιτήσεις του διαχειριστή. Στον κατάλογο

```

/usr/share/openstack-dashboard/openstack_dashboard/static/dashboard/

```

υπάρχουν τα αρχεία μορφοποίησης της διεπαφής όπως css, εικόνες, frames και templates και στην θέση

```

/usr/share/openstack-dashboard/openstack_dashboard/templates

```

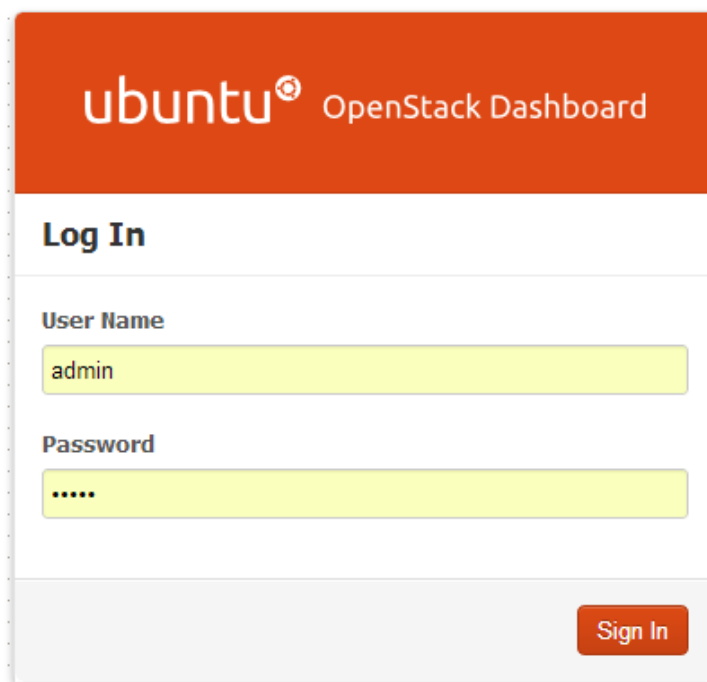
τα αρχεία html. Τέλος, η ανάπτυξη γίνεται στο αρχείο /etc/openstack-dashboard/local_settings.py .

7. Διεπαφή Χρήστη (Dashboard)

Χρησιμοποιώντας το Dashboard ο χρήστης μπορεί να ελέγχει ορισμένες υπηρεσίες του Openstack. Μπορεί να χρησιμοποιηθεί για διαχείριση των instances και των εικόνων, δημιουργία Keypairs, Containers κ.λ.π.

Είσοδος

Είσοδος στο Dashboard με όνομα χρήστη “admin” και κωδικό “admin”.



The image shows a login form for the Ubuntu OpenStack Dashboard. The form has an orange header with the 'ubuntu' logo and 'OpenStack Dashboard' text. Below the header, the title 'Log In' is displayed. There are two input fields: 'User Name' with the value 'admin' and 'Password' with masked characters '.....'. A red 'Sign In' button is located at the bottom right of the form.

ubuntu[®] OpenStack Dashboard

Log In

User Name
admin

Password
.....

Sign In

Instances και Volumes

Εμφανίζονται τα διαθέσιμα instances και volumes που έχουν δημιουργηθεί για το tenant στο οποίο ανήκει ο χρήστης που έχει εισέλθει. Υπάρχει επίσης η δυνατότητα δημιουργίας και τερματισμού instances και volumes. Ο όρος project αναφέρεται στο tenant.

Instances & Volumes

Instances

<input type="checkbox"/>	Instance Name	IP Address	Size	Status	Task	Power State	Actions
<input type="checkbox"/>	instance	192.168.0.34 195.251.209.14	4GB RAM 2 VCPU 10.0GB Disk	Hard Reboot	Rebooting Hard	Running	Edit Instance

Displaying 1 item

Volumes

<input type="checkbox"/>	Name	Description	Size	Status	Attachments	Actions
<input type="checkbox"/>	myvolume	-	2 GB	Available	-	Edit Attachments

Displaying 1 item

Κονσόλα VNC

Η σύνδεση σε ένα ενεργό instance γίνεται μέσω της κονσόλας VNC.

Project

Admin

PROJECT

admin

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Instance Detail: myinstance

Overview

Log

VNC

Instance VNC Console

If VNC console is not responding to keyboard input, click the grey status bar below.

Connected (unencrypted) to: QEMU (instance-00000006)

Send Ctrl+Alt+Del

```

Ubuntu 12.10 myinstance tty1
myinstance login: _

```

193.251.209.0/nova/instances_and_volumes/

Project

Admin

PROJECT

admin

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Instance Detail: myinstance

Overview

Log

VNC

Instance VNC Console

If VNC console is not responding to keyboard input, click the grey status bar below.

Connected (unencrypted) to: QEMU (instance-00000006)

Send Ctrl+Alt+Del

```

Ubuntu 12.10 myinstance tty1
myinstance login: root
Last login: Fri Nov  2 14:17:00 UTC 2012 on tty1
Welcome to Ubuntu 12.10 (GNU/Linux 3.5.0-13-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Fri Nov  2 14:34:23 UTC 2012

System load:  0.95      Processes:    68
Usage of /:   42.9% of 1.96GB   Users logged in:  0
Memory usage: 7%        IP address for eth0: 192.168.0.34
Swap usage:  0%

Graph this data and manage this system at https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest
http://www.ubuntu.com/business/services/cloud
root@myinstance:~# _

```

193.251.209.0/nova/instances_and_volumes/

Images και Snapshots

Περιέχει τις διάφορες εικόνες που έχουν φορτωθεί. Από εδώ μπορούμε να επεξεργαστούμε τις εικόνες και να διαγράψουμε ή να δημιουργήσουμε στιγμιότυπα.

The screenshot shows the Ubuntu OpenStack Dashboard interface. The top navigation bar is orange with the 'ubuntu' logo and 'OpenStack Dashboard' text. On the right, it says 'Logged in as: admin' with links for 'Settings' and 'Sign Out'. The left sidebar contains a 'Project' dropdown set to 'admin' and a menu with options: 'Manage Compute' (Overview, Instances & Volumes), 'Images & Snapshots' (selected), 'Access & Security', 'Object Store', and 'Containers'. The main content area is titled 'Images & Snapshots' and contains three sections: 'Images', 'Instance Snapshots', and 'Volume Snapshots'. The 'Images' section has a 'Delete Images' button and a table with one item: 'Ubuntu Server test' (Image, Active, Yes, OVF) with a 'Launch' button. The 'Instance Snapshots' and 'Volume Snapshots' sections both show 'No items to display'.

Image Name	Type	Status	Public	Container Format	Actions
Ubuntu Server test	Image	Active	Yes	OVF	Launch

Image Name	Type	Status	Public	Container Format	Actions
No items to display.					

Name	Description	Size	Status	Volume ID	Actions
No items to display.					

Access και Security

Από εδώ γίνεται η διαχείριση των floating IP's που συνδέονται σε ένα στιγμιότυπο ή απελευθερώνονται από αυτό. Μπορούν να δημιουργηθούν επίσης ομάδες ασφαλείας.

ProjectAdmin

PROJECTadmin

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Access & Security

Floating IPs

Allocate IP To ProjectRelease Floating IPs

<input type="checkbox"/>	IP Address	Instance	Floating IP Pool	Actions
<input type="checkbox"/>	195.251.209.14	625f8c25-2342-4cf2-9d83-8305c65efc43	nova	Disassociate IP

Displaying 1 item

Security Groups

Create Security GroupDelete Security Groups

<input type="checkbox"/>	Name	Description	Actions
<input type="checkbox"/>	default	default	Edit Rules
<input type="checkbox"/>	mysecurity	test	Edit Rules

Displaying 2 items

Keypairs

Create KeypairImport KeypairDelete Keypairs

<input type="checkbox"/>	Keypair Name	Fingerprint	Actions
<input type="checkbox"/>	mykey	fd:65:68:89:cc:d9:d8:9f:62:57:47:a9:be:8e:63:56	Delete Keypair

Containers και Objects

Σε αυτή την σελίδα δημιουργούνται ή διαγράφονται containers. Επίσης φορτώνονται ή μεταφορτώνονται αντικείμενα.

ubuntu[®] OpenStack Dashboard

Logged in as: adminSettingsSign Out

ProjectAdmin

PROJECTadmin

Manage Compute

Overview

Instances & Volumes

Images & Snapshots

Access & Security

Object Store

Containers

Containers

Containers

Create ContainerDelete Containers

<input type="checkbox"/>	Container Name	Objects	Size	Actions
<input type="checkbox"/>	test1	1	76.5 KB	List Objects
<input type="checkbox"/>	test2	0	0 bytes	List Objects

Displaying 2 items

Βιβλιογραφία

Βιβλία

Anthony T.Velte, Toby J. Velte, Robert Elsenpeter (2010): Cloud Computing A Practical Approach

Ιστοσελίδες

<http://wiki.openstack.org/>

http://www.linux-kvm.org/page/Main_Page

<http://searchdatacenter.techtarget.com/definition/back-end>

<http://pythonpaste.org/deploy/>

http://www.linux-kvm.org/page/Main_Page

http://foss.ntua.gr/wiki/index.php/TCP/IP_Basics#Internet_Protocol_.CE.BA.CE.B1.CE.B9_.CE.B4.CE.B9.CE.B5.CF.85.CE.B8.CF.85.CE.BD.CF.83.CE.B9.CE.BF.CE.B4.CF.8C.CF.84.CE.B7.CF.83.CE.B7

<http://www.mirantis.com/blog/configuring-floating-ip-addresses-networking-openstack-public-private-clouds/>

<http://docs.openstack.org/diablo/openstack-compute/admin/content/allocating-associating-ip-addresses.html>