

Τ.Ε.Ι. ΠΕΙΡΑΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Εργαστήριο Λειτουργικών Συστημάτων

Τίτλος Πτυχιακής Εργασίας

Υλοποίηση OpenStack

σε

Εικονικό Περιβάλλον

Σπουδαστές: Βουρόπουλος Θεόδωρος | Πυραλής Πάρις-Γεώργιος

Εισηγητές: Γεώργιος Διλιντάς | Στυλιανός Βουτσινάς



Πειραιάς

Έτος 2014

Περιεχόμενα

Ευρετήριο εικόνων	6
Εισαγωγή	7
Cloud Computing	8
<i>Ορισμός του cloud computing.....</i>	<i>8</i>
<i>Μοντέλα ανάπτυξης Cloud Computing.....</i>	<i>9</i>
Public cloud.....	9
Ορισμός.....	9
Private cloud.....	9
Ορισμός.....	9
Community cloud.....	10
Ορισμός.....	10
Hybrid cloud.....	10
Ορισμός.....	10
<i>Μοντέλα υπηρεσιών Cloud Computing</i>	<i>10</i>
IaaS	10
Ορισμός.....	10
Ανάλυση.....	10
PaaS	11
Ορισμός.....	11
Ανάλυση.....	11
SaaS	12
Ορισμός.....	12
Ανάλυση.....	12
Hardware Virtualization	13
Οδηγός έναρξης για το Openstack Compute.....	15
<i>Εισαγωγή.....</i>	<i>15</i>
<i>Αρχιτεκτονική του OpenStack</i>	<i>15</i>
Horizon (Dashboard).....	15
Nova (Compute).....	15
Neutron (Networking).....	15
Swift (Object Storage).....	15

Cinder (Block Storage)	15
Keystone (Identity Service).....	15
Glance (Image Service).....	15
Ceilometer (Telemetry).....	15
<i>Εγκατάσταση και Διαμόρφωση</i>	16
Server1	17
Base OS.....	17
Ρύθμιση του Δικτύου	17
NTP Server.....	18
Βάσεις δεδομένων.....	18
MySQL	19
Δημιουργία Βάσεων Δεδομένων	19
Keystone	20
Δημιουργία Tenants.....	21
Δημιουργία χρηστών	21
Δημιουργία ρόλων-ROLES	21
Εμφάνιση σε λίστα των Tenants, Users και Roles.....	21
Προσθήκη ρόλων σε Tenants	22
Δημιουργία Services.....	23
Δημιουργώντας Endpoints.....	24
Glance	24
Ρύθμιση Glance	24
Nova.....	25
Nova Configuration	26
OpenStack Dashboard.....	28
Swift.....	29
Swift Installation.....	29
Swift Storage Backends.....	29
Ρύθμιση Rsync.....	31
Ρύθμιση Swift Components.....	33
Starting Swift services	40
Testing Swift.....	40
Server2	41
Λειτουργικό Σύστημα	41
Ρύθμιση του Δικτύου	41
NTP Client	41

Nova Components (nova-compute μόνο).....	42
ClientI	44
Βασικό ΛΣ	44
Ρύθμιση του Δικτύου	44
NTP Client	44
Client Tools.....	44
OpenStack Dashboard.....	45
<i>Διαχείριση Image</i>	45
Εισαγωγή.....	45
Δημιουργία ενός Linux Image.....	46
Εγκατάσταση ΛΣ	46
Ubuntu	47
Fedora.....	47
CentOS 6 and RHEL 6	48
Uploading the Linux image.....	48
Δημιουργία ενός Windows Image	48
Εγκατάσταση ΛΣ	49
Uploading the Windows image.....	49
<i>Διαχείριση Instance</i>	50
Εισαγωγή.....	50
Openstack Command Line Tools.....	50
Creation of Key Pairs.....	50
Εκκίνηση και Διαχείριση instances.....	51
<i>OpenStack Dashboard (Horizon)</i>	52
Login	52
Επισκόπηση χρήστη	53
Instances.....	53
Services.....	54
Flavors.....	54
Images.....	55
Projects.....	55
Users	56
Quotas	56
Επισκόπηση Project.....	57
Instances & Volumes	57
Instances - VNC Console	58

Images & Snapshots	58
Access & Security	59
Containers & Objects	61
<i>Διαχείριση Storage</i>	62
Nova-volume	62
Αλληλεπίδραση με τον ελεγκτή αποθήκευσης	62
Swift	63
<i>Διαχείριση Δικτύου</i>	66
Εισαγωγή	66
<i>Ασφάλεια</i>	67
Επισκόπηση Ασφαλείας	67
<i>OpenStack Commands</i>	68
Nova Commands	68
Glance Commands	69
Swift Commands	70
Keystone Commands	70
Υλοποίηση	71
Εκτίμηση Αποτελεσμάτων	73
Βιβλιογραφία	74
Παραρτήματα	77
<i>Παράρτημα Α</i>	77
Πίνακας εικόνων	77
Εισαγωγή στο OpenStack και στα Components του	77

Ευρετήριο εικόνων

Εικ 1	Σύγκριση παραδοσιακών δομών και υπηρεσιών cloud	12
Εικ 2	Εταιρίες που δραστηριοποιούνται στο Cloud	13
Εικ 3	Παραδειγμα Virtualisation	14
Εικ 4	Στησιμο OpenStack	16
Εικ 5	Σελίδα Login	52
Εικ 6	Σελίδα Επισκοπισης	53
Εικ 7	Σελίδα Στιγμιοτυπων	53
Εικ 8	Σελίδα Υπηρεσιων	54
Εικ 9	Σελίδα Flavors	54
Εικ 10	Σελίδα Images	55
Εικ 11	Σελίδα Projects.....	55
Εικ 12	Σελίδα Χρηστων	56
Εικ 13	Σελίδα Αριθμου διαθεσιμων πορων	56
Εικ 14	Επισκοπιση Project.....	57
Εικ 15	Σελίδα Στιγμιοτυπων και Τομων	57
Εικ 16	Σελίδα Κονσολας VNC Στιγμιοτυπου	58
Εικ 17	Σελίδα Images και Snapshots.....	59
Εικ 18	Σελίδα Προσβασης και Ασφαλειας	60
Εικ 19	Σελίδα δημιουργιας Κανονων Ασφαλειας	60
Εικ 20	Σελίδα δημιουργιας Keypair	61
Εικ 21	Σελίδα ελεγχου Containers	61
Εικ 22	Σελίδα ελεγχου Objects.....	62

Εισαγωγή

Σκοπός αυτής της εργασίας είναι η δημιουργία μια υπηρεσίας η οποία θα δίνει την δυνατότητα ελέγχου υπολογιστικών, αποθηκευτικών και δικτυακών πόρων μέσω ενός Web Browser μέσω της εφαρμογής cloud computing OpenStack. Το OpenStack κυκλοφόρησε τον Οκτώβρη του 2010 και είναι γραμμένο σε Python. Ανήκει στην κατηγορία υπηρεσιών Iaas (Infrastructure as a Service) του Cloud Computing. Το project ξεκίνησαν από κοινού η NASA και η Rackspace Hosting. [1] Υποστηρίζεται από πλήθος εταιρών και οργανισμών όπως οι Arista Networks, AT&T, AMD, Brocade Communications Systems, Canonical, Cisco, Dell, EMC, Ericsson, F5 Networks, Go Daddy, Groupe Bull, Hewlett-Packard, IBM, Inktank, Intel, NEC, NetApp, Nexenta, Rackspace Hosting, Red Hat, SUSE Linux, VMware, Oracle and Yahoo!. κ.α. με ένα συνολικό αριθμό που ξεπερνά τις 300. [2]

Επίσης θα αναλυθούν βασικές έννοιες του cloud computing, πως ορίζεται και τι υπηρεσίες περιλαμβάνει, ώστε να μπορέσει να αντιληφθεί ο αναγνώστης το πλαίσιο που έχει διαμορφωθεί και τι θέση έχει το Openstack σαν υπηρεσία μέσα του.

Θα αναλυθεί η διαδικασία που ακολουθήθηκε και το σκεπτικό με το οποίο επιλέχθηκε το Openstack για την υλοποίηση του project και θα γίνει μια ανάλυση του σχεδιασμού του Openstack και των στοιχείων από τα οποία αποτελείται. Επίσης θα αναλυθεί η διαδικασία εγκατάστασης του και θα δοθούν οδηγίες για την σωστή χρήση του.

Cloud Computing

Ορισμός του cloud computing

Το cloud computing είναι ένα μοντέλο για τη παροχή δυνατότητας πανταχού παρούσας, βολικής, κατά παραγγελία πρόσβασης στο δίκτυο σε κοινόχρηστη ομάδα διαμορφώσιμων υπολογιστικών πόρων (π.χ., δίκτυα, servers, storage, εφαρμογές και υπηρεσίες) που μπορούν να αποδοθούν γρήγορα σε χρήση με την ελάχιστη προσπάθεια διαχείρισης ή αλληλεπίδρασης από τους παρόχους υπηρεσιών. Αυτό το μοντέλο σύννεφου αποτελείται από πέντε βασικά χαρακτηριστικά, τρία μοντέλα υπηρεσιών, και τέσσερα μοντέλα ανάπτυξης [3]

Απαραίτητα Χαρακτηριστικά

Αυτοεξυπηρέτηση κατά απαίτηση. (On-demand self-service.)

Ένας καταναλωτής μπορεί μονομερώς να προμηθευτεί δυνατότητες αξιοποίησης της υπολογιστικής διάταξης, όπως είναι ο χρόνος διακομιστή και αποθήκευσης δικτύου, ανάλογα με τις ανάγκες αυτομάτως χωρίς να απαιτείται ανθρώπινη αλληλεπίδραση με κάθε φορά παροχής υπηρεσιών.

Ευρεία πρόσβαση στο δίκτυο.(Broad network access.)

Δυνατότητες είναι διαθέσιμες μέσω του δικτύου και προσβάσιμες μέσω τυποποιημένων μηχανισμών που προωθούν την χρήση από ετερογενείς πλατφόρμες thin ή thick client (π.χ., κινητά τηλέφωνα, ταμπλέτες, φορητούς υπολογιστές, και workstations).

Συγκέντρωση πόρων.(Resource pooling.)

Οι υπολογιστικοί πόροι του παρόχου συγκεντρώνονται για να εξυπηρετήσει πολλαπλούς καταναλωτές χρησιμοποιώντας ένα μοντέλο πολλαπλών ενοικιαστών, με διαφορετικούς φυσικούς και εικονικούς πόρους που έχουν εκχωρηθεί δυναμικά και αναδιατίθενται ανάλογα με την ζήτηση των καταναλωτών. Υπάρχει μια αίσθηση ανεξαρτησίας τοποθεσίας στο ότι ο πελάτης δεν έχει γενικά έλεγχο ή γνώση πάνω από την ακριβή τοποθεσία των παρεχόμενων πόρων, αλλά μπορεί να είναι σε θέση να προσδιορίζει τη θέση σε υψηλότερο επίπεδο αφαίρεσης (π.χ., χώρα, κράτος, ή κέντρο δεδομένων). Παραδείγματα πόρων περιλαμβάνουν την αποθήκευση, την επεξεργασία, τη μνήμη και το εύρος ζώνης δικτύου.

Ταχεία ελαστικότητα.(Rapid elasticity.)

Δυνατότητες μπορεί να αποδοθούν και να απελευθερωθούν ελαστικά, σε ορισμένες περιπτώσεις, αυτόματα, να κλιμακωθούν γρήγορα προς τα έξω και προς τα μέσα ανάλογα με τη

ζήτηση. Για τον καταναλωτή, οι δυνατότητες που διατίθενται συχνά φαίνεται να είναι απεριόριστες και μπορούν να διατεθούν σε οποιαδήποτε ποσότητα, ανά πάσα στιγμή.

Μετρούμενες υπηρεσίες.(Measured service.)

Τα συστήματα cloud ελέγχουν αυτόματα τη βέλτιστη χρήση των πόρων αξιοποιώντας μια δυνατότητα μέτρησης σε κάποιο επίπεδο αφαίρεσης ανάλογα με το είδος της υπηρεσίας (π.χ., αποθήκευση, επεξεργασία, εύρος ζώνης, και ενεργούς λογαριασμούς χρηστών). Η χρήση των πόρων μπορεί να παρακολουθείται, ελέγχεται, και να αναφέρεται, παρέχοντας διαφάνεια τόσο για τον πάροχο όσο και για τον καταναλωτή της χρησιμοποιούμενης υπηρεσίας. [3]

Μοντέλα ανάπτυξης Cloud Computing

Public cloud

Ορισμός

Η υποδομή cloud τροφοδοτείται για ελεύθερη χρήση από το ευρύ κοινό. Μπορεί να ανήκει, διοικείται, και λειτουργεί από μια επιχείρηση, ακαδημαϊκή ή κυβερνητική οργάνωση, ή κάποιο συνδυασμό αυτών. Υπάρχει στις εγκαταστάσεις του φορέα παροχής cloud. [3]

Παραδείγματα υπηρεσιών public cloud είναι το Email που προσφέρεται από εταιρείες όπως οι Microsoft, Yahoo και Google [4], υπηρεσίες αποθήκευσης δεδομένων όπως το Dropbox [5] καθώς και το Amazon Elastic Compute Cloud (EC2) [6]

Private cloud

Ορισμός

Η υποδομή cloud τροφοδοτείται για αποκλειστική χρήση από έναν μόνο φορέα που περιλαμβάνει πολλαπλούς καταναλωτές (π.χ., επιχειρηματικές μονάδες). Μπορεί να ανήκει, διοικείται, και λειτουργεί από τον ίδιο τον φορέα, έναν τρίτο, ή κάποιο συνδυασμό αυτών, και μπορεί να υπάρχει εντός ή εκτός των χώρων του φορέα. [3]

Κάποιες από τις εταιρείες που προσφέρουν λύσεις για private cloud είναι η Cisco [7], η Dell [8], η IBM [9].

Community cloud

Ορισμός

Η υποδομή cloud τροφοδοτείται για την αποκλειστική χρήση από μια συγκεκριμένη κοινότητα καταναλωτών οργανισμών που έχουν κοινές ανησυχίες (π.χ. αποστολή, απαιτήσεις ασφάλειας, πολιτικής, και κριτήρια συμμόρφωσης). Μπορεί να ανήκει, διοικείται, και λειτουργεί από ένα ή περισσότερους από τους οργανισμούς στην κοινότητα, έναν τρίτο, ή κάποιο συνδυασμό αυτών, και μπορεί να υπάρχει εντός ή εκτός των χώρων του φορέα. [3]

Hybrid cloud

Ορισμός

Η υποδομή cloud είναι μια σύνθεση από δύο ή περισσότερες διακριτών υποδομών cloud (ιδιωτικών, κοινοτικών ή δημόσιων) που παραμένουν μοναδικές οντότητες, αλλά συνδέονται με προτυποποιημένη ή αποκλειστική τεχνολογία που επιτρέπει την φορητότητα δεδομένων και εφαρμογών (π.χ. Cloud bursting για την εξισορρόπηση φορτίου μεταξύ δυο clouds). [3]

Μοντέλα υπηρεσιών Cloud Computing

Iaas

Ορισμός

Infrastructure as a Service. Η δυνατότητα που παρέχεται στον καταναλωτή είναι η παροχή επεξεργαστικής ισχύος, αποθηκευτικού χώρου, δικτύων και άλλων θεμελιωδών υπολογιστικών πόρων, όπου ο καταναλωτής είναι σε θέση να αναπτύξει και να εκτελέσει οποιοδήποτε λογισμικό, το οποίο μπορεί να περιλαμβάνει λειτουργικά συστήματα και εφαρμογές. Ο καταναλωτής δεν διαχειρίζεται ή έχει τον έλεγχο της υποκείμενης υποδομής cloud, αλλά έχει τον έλεγχο των λειτουργικών συστημάτων, του αποθηκευτικού χώρου και των εγκατεστημένων εφαρμογών, και ενδεχομένως περιορισμένο έλεγχο επιλεγμένων στοιχείων δικτύωσης (π.χ. Firewalls). [3]

Ανάλυση

Ξεκινώντας από το φυσικό επίπεδο το πρώτο μοντέλο υπηρεσιών cloud είναι το Infrastructure as a Service. Είναι η χαμηλότερου επιπέδου υπηρεσία που προσφέρεται και δίνει ελεγχόμενη πρόσβαση σε μια εικονική υποδομή που περιλαμβάνει server, δίκτυα και αποθηκευτικό χώρο όπου μπορούν να αναπτυχθούν λειτουργικά συστήματα και εφαρμογές.

Μπορεί να χρησιμοποιηθεί είτε για επέκταση του υπάρχοντος υλικού μιας επιχείρησης, είτε για αποφυγή αγοράς νέου εξοπλισμού. Ο πελάτης έχει τον έλεγχο του λογισμικού, χωρίς να νοιάζεται για το υλικό. Αν και οι πόροι είναι εικονικοί, ο πελάτης δεν το αντιλαμβάνεται και μπορούν να δοθούν στον πελάτη με ή χωρίς λογισμικό. Το κόστος της υπηρεσίας καθορίζεται

από την χρήση του συστήματος από τον πελάτη σύμφωνα με τις ανάγκες του. Αν αυξηθούν οι απαιτήσεις μπορεί ανά πάσα στιγμή να ζητήσει περισσότερους πόρους, ώστε να ικανοποιήσει τις ανάγκες του καθώς και το αντίστροφο. [10] Ανάμεσα στους μεγαλύτερους προμηθευτές IaaS είναι οι IBM, Amazon Web Services, Bluelock, CSC και Rackspace. [11] [12] [13] [14] [15] Χαρακτηριστικά παραδείγματα IaaS είναι EC2 της Amazon και το OpenStack.

Paas

Ορισμός

Platform as a Service. Η δυνατότητα που παρέχεται στον καταναλωτή είναι να αναπτύξει πάνω στις υποδομές cloud δικές του εφαρμογές ή εφαρμογές που έχουν δημιουργηθεί με τη χρήση γλωσσών προγραμματισμού, βιβλιοθηκών, υπηρεσιών και εργαλείων που υποστηρίζονται από τον πάροχο. Ο καταναλωτής δεν διαχειρίζεται ή ελέγχει την σχετική υποδομή cloud, συμπεριλαμβανομένου του δικτύου, των servers, των λειτουργικών συστημάτων και των αποθηκευτικών μέσων, αλλά έχει τον έλεγχο των χρησιμοποιημένων εφαρμογών και ενδεχομένως των ρυθμίσεων διαμόρφωσης του περιβάλλοντος που φιλοξενεί την εφαρμογή. [3]

Ανάλυση

Το μοντέλο Platform as a Service (PaaS) στήνεται πάνω στο IaaS. Είναι έτοιμο ώστε να στηθούν πάνω του εφαρμογές, καθώς το λειτουργικό σύστημα και διάφορα εργαλεία όπως οι μεταγλωττιστές είναι ήδη εγκατεστημένα και η διαχείριση τους γίνεται από τον πάροχο. Ο πελάτης μπορεί να εμπλουτίσει τα διάφορα εργαλεία, αλλά ο έλεγχος της πλατφόρμας παραμένει στον πάροχο.

Το PaaS μπορεί να παραλληλιστεί με το παραδοσιακό web hosting, με την μεγάλη διαφορά στην ελαστικότητα των υπηρεσιών, γιατί είναι μια σουίτα εργαλείων που επιτρέπει την δημιουργία και διαχείριση εφαρμογών στο cloud. Περιέχει εργαλεία για ανάπτυξη, δοκιμή αλλά και ένα περιβάλλον εκτέλεσης εφαρμογών. Όπως και στο IaaS, οι πόροι είναι δυναμικά μεταβαλλόμενοι, και η διαχείριση τους γίνεται από τον πάροχο χωρίς να επιβαρύνεται ο πελάτης με επιπλέον ευθύνες. Ένα από τα πιθανά μειονεκτήματα είναι η έλλειψη φορητότητας και η εξάρτηση από ένα πάροχο καθώς η ανάπτυξη εφαρμογών γίνεται πάνω σε δικά του εργαλεία. Είναι όμως καλή επιλογή αν το περιβάλλον ανάπτυξης εφαρμογών του πελάτη είναι ίδιο με αυτό του παρόχου PaaS ή για δοκιμή εφαρμογών και υπηρεσιών που μπορούν να αναπτυχθούν γρήγορα από της ήδη υπάρχουσες υπηρεσίες που προσφέρει η πλατφόρμα. [10]

Χαρακτηριστικά παραδείγματα του μοντέλου PaaS είναι τα AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com και Google App Engine. [16] [17] [18] [19] [20]

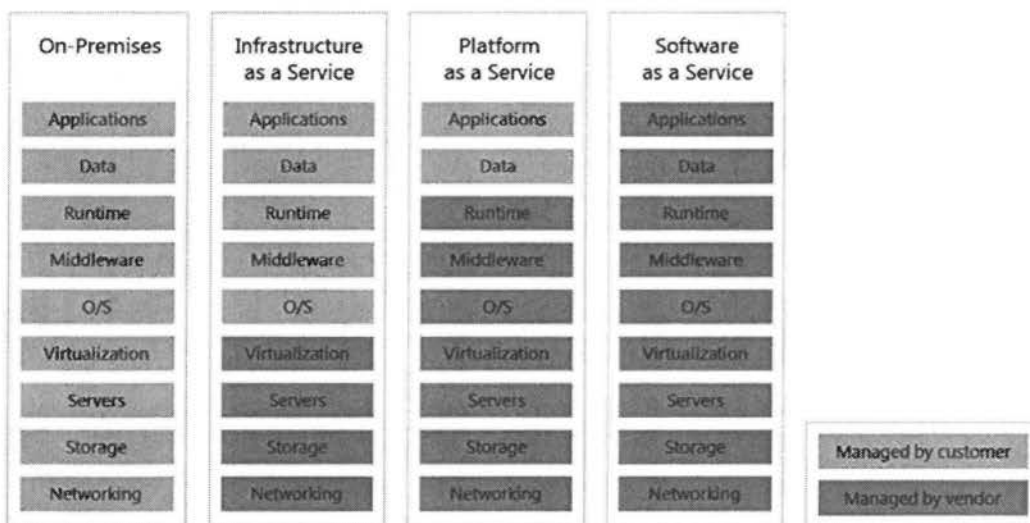
Saas

Ορισμός

Software as a Service. Η δυνατότητα που δίνεται στον χρήστη είναι η χρήση των εφαρμογών του παρόχου που τρέχουν πάνω σε μια δομή cloud. Οι εφαρμογές είναι διαθέσιμες μέσω ενός web interface ή μέσω ενός προγράμματος. Ο χρήστης δεν διαχειρίζεται ή ελέγχει την υποδομή cloud σε κανένα επίπεδο, με πιθανή εξαίρεση κάποιων περιορισμένων ρυθμίσεων, της εφαρμογής, που αφορούν τον χρήστη. [3]

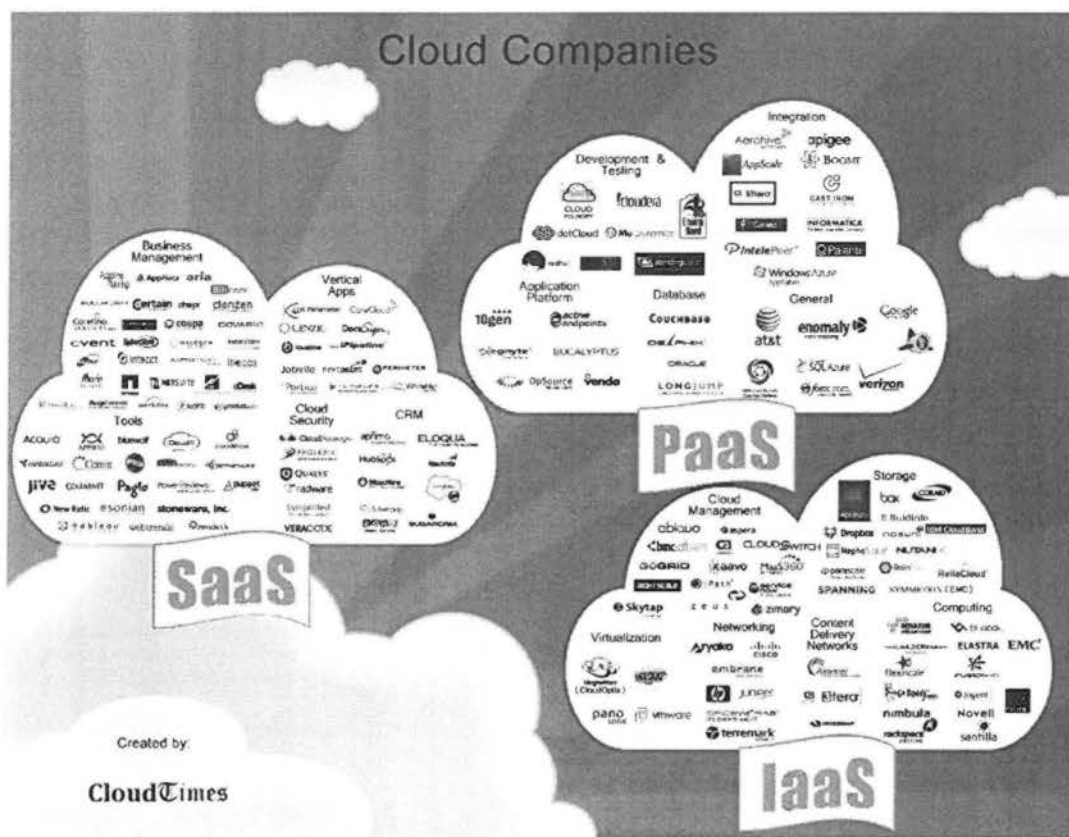
Ανάλυση

Το Software as a Service (SaaS) είναι ένα μοντέλο υπηρεσιών αφαιρεί από τον χρήστη την ευθύνη διαχείρισης της υποδομής cloud, δίνοντας βάση στο επίπεδο εφαρμογής. Οι εφαρμογές είναι διαθέσιμες μέσω ενός web interface ή μέσω ενός προγράμματος όπως μια εφαρμογή για κινητά τηλέφωνα. Το Gmail της Google είναι το πιο δημοφιλές παράδειγμα εφαρμογής cloud computing. Μια εταιρεία μπορεί να χρησιμοποιήσει το Gmail χωρίς να νοιαστεί για την συντήρηση του υλικού, την ασφάλεια, το uptime και την διαχείριση της υποδομής. Ο χρήστης μπορεί να ρυθμίσει παραμέτρους του προγράμματος, οι οποίες όμως γίνονται μέσα από το interface της εφαρμογής. Έτσι ο χρήστης χρησιμοποιεί μια υπηρεσία email χωρίς να τον ενδιαφέρει πως παρέχεται αυτή. Είναι ο ευκολότερος τρόπος για την είσοδο στον κόσμο του cloud computing. Χαρακτηριστικά παραδείγματα SaaS είναι το Gmail της Google, καθώς και το Dropbox. [21] [5]



Εικ 1 Σύγκριση παραδοσιακών δομών και υπηρεσιών cloud¹

¹ <http://venturebeat.com/2011/11/14/cloud-iaas-paas-saas/>



Εικ 2 Εταιρίες που δραστηριοποιούνται στο Cloud²

Hardware Virtualization

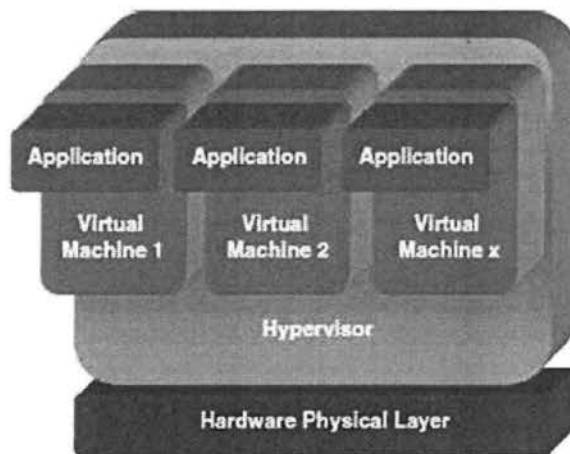
Το Hardware virtualization είναι μια αναπτυσσόμενη τεχνολογία που αξιοποιεί την αύξηση στην επεξεργαστική ισχύ επιτρέποντας την δημιουργία εικονικών μηχανών για την προσομοίωση φυσικής υποδομής. Αυτή η τεχνολογία έχει επιτρέψει σε εταιρείες όπως τα data centers να βελτιώσουν την χρήση και διαχείριση των πόρων τους χτίζοντας εικονικά επίπεδα υλικού στα φυσικά μηχανήματα που έχουν. Τα εικονικά επίπεδα επιτρέπουν στους διαχειριστές των data centers να δημιουργούν νέα instances εικονικού υλικού ανεξάρτητα από τις συσκευές που υπάρχουν από κάτω. Επίσης νέο υλικό μπορεί να προστεθεί στο ήδη υπάρχων, με αποτέλεσμα την αύξηση της υπολογιστικής ισχύος, της χωρητικότητας και της μνήμης χωρίς να επηρεαστεί το εικονικό επίπεδο παρά μόνο στην αύξηση των δυνατοτήτων του.

Στην εικόνα 3 φαίνονται τα κυριότερα μέρη μιας εικονικής αρχιτεκτονικής. Ξεκινώντας από το φυσικό επίπεδο πρώτα υπάρχει ο hypervisor. Ο ρόλος του είναι να προσφέρει ένα τρόπο με τον οποίο οι εικονικές μηχανές θα επικοινωνούν με το φυσικό επίπεδο χωρίς την

² <http://cloudtimes.org/2011/11/30/top-paas-saas-and-iaas-cloud-companies-by-cloudtimes/>

εγκατάσταση λειτουργικού συστήματος. Οι εικονικές μηχανές εγκαθίστανται πάνω σε αυτόν. Κάθε εικονική μηχανή παρουσιάζεται σαν ένας διακριτός πόρος, χωρίς όμως να υπάρχει στην πραγματικότητα. Ένα λειτουργικό σύστημα εγκαθίσταται πάνω στις εικονικές μηχανές, δίνοντας την δυνατότητα να εκτελεστούν πάνω του παραδοσιακές εφαρμογές.

Το virtualization προσφέρει τρία βασικά πλεονεκτήματα στην διαχείριση ενός data center. Πρώτον, οι εφαρμογές είναι περιορισμένες σε μια εικονική μηχανή, κάτι το οποίο αυξάνει την ασφάλεια και απομονώνει οποιαδήποτε επιζήμια ενέργεια στην απόδοση της μηχανής από το υπόλοιπο σύστημα. Δεύτερον, η ενοποίηση ανόμοιων πλατφορμών σε ένα ενιαίο επίπεδο υλικού σημαίνει ότι μπορεί να γίνει καλύτερη χρήση του συστήματος, οδηγώντας σε αυξημένη απόδοση. Τρίτον, επιτρέπει την αποθήκευση guest λειτουργικών συστημάτων σαν snapshots που διατηρούν τις προϋπάρχουσες ρυθμίσεις τους, ώστε να είναι δυνατή η γρήγορη επαναφορά με περίπτωση βλάβης.



Εικ 3 Παράδειγμα Virtualization³

³ OpenStack Beginner's Guide (for Ubuntu - Precise) v3.0

Οδηγός έναρξης για το Openstack Compute

Εισαγωγή

Αρχιτεκτονική του OpenStack

Το OpenStack αποτελείται από 10 Projects που το καθένα διαχειρίζεται μια υπηρεσία.

Horizon (Dashboard)

Παρέχει ένα web-based portal που αλληλοεπιδρά με τις υποκείμενες υπηρεσίες του OpenStack, όπως η εκκίνηση ενός instance, η ανάθεση IP και οι ρυθμίσεις του ελέγχου πρόσβασης.

Nova (Compute)

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

Neutron (Networking)

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

Swift (Object Storage)

Αποθηκεύει και ανακτά μη δομημένα δεδομένα μέσω ενός API.

Cinder (Block Storage)

Παρέχει επίμονη αποθήκευση για block αρχεία που χρησιμοποιούνται για την λειτουργία των instances. Μέσω αυτού γίνεται η δημιουργία και η διαχείριση συσκευών block storage.

Keystone (Identity Service)

Παρέχει την υπηρεσία πιστοποίησης και εξουσιοδότησης στις υπόλοιπες υπηρεσίες.

Glance (Image Service)

Αποθηκεύει και ανακτά virtual machine disk images που χρησιμοποιεί το OpenStack κατά την δημιουργία instances.

Ceilometer (Telemetry)

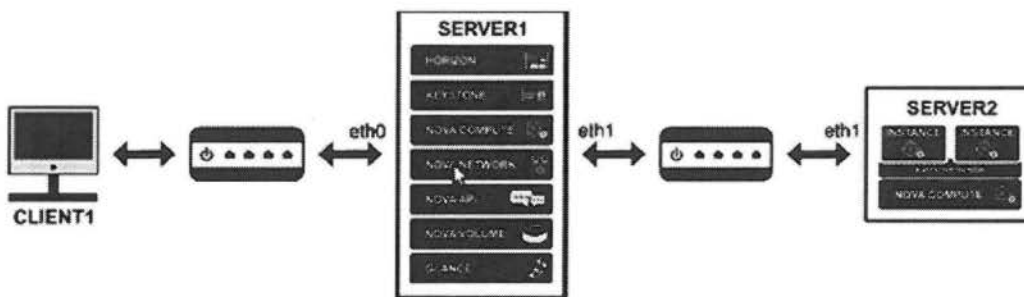
Επιτηρεί και μετρά το OpenStack για benchmarking, επεκτασιμότητα, στατιστικούς σκοπούς και χρεώσεις.

Εγκατάσταση και Διαμόρφωση

Η ακόλουθη ενότητα περιγράφει πώς δημιουργείται μια ελάχιστη υποδομή cloud που βασίζεται στο OpenStack χρησιμοποιώντας 3 μηχανήματα.

Τα μηχανήματα που αναφέρονται στο παρόν και στα επόμενα κεφάλαια, είναι οι Server1, Server2 και Client1 . Ο Server1 τρέχει όλα τα στοιχεία της Nova, Glance, Swift, Keystone και Horizon (GUI Πλατφόρμα Openstack). Ο Server2 τρέχει μόνο nova – compute. Στο Openstack όλα τα συστατικά ακολουθούν μια πολιτική τέτοια ώστε, κάθε εξάρτημα ή κάθε ομάδα συστατικών μπορεί να εγκατασταθεί σε οποιονδήποτε διακομιστή .

Ο Client1 δεν είναι ένα απαιτούμενο στοιχείο. Στην εγκατάσταση του δείγματος μας, χρησιμοποιείται για την ομαδοποίηση εικόνων-images , ως client για το web interface και να τρέξει εντολές Openstack για τη διαχείριση της υποδομής . Έχοντας τον client προσομοιάζουμε πως θα διαχειριζόταν κάποιος απομακρυσμένα την υποδομή. Επίσης , ο συνδυασμός των desktop συστημάτων, συμπεριλαμβανομένων των Windows, θα χρειαστούν ένα γραφικό περιβάλλον και είναι καλύτερα να έχουν ένα ειδικό μηχάνημα για το σκοπό αυτό. Θα συνιστούσαμε αυτό το μηχάνημα να είναι VT- Enabled ώστε να τρέχει το KVM το οποίο μπορεί να δημιουργεί εικονικές μηχανές κατά τη δημιουργία της εικόνας-image για την ομαδοποίηση.



Εικ 4 Στήσιμο OpenStack⁴

⁴ OpenStack Beginner's Guide (for Ubuntu - Precise) v3.0

Server1

Όπως φαίνεται στο παραπάνω σχήμα, ο Server1 περιέχει όλα τα nova-services, συμπεριλαμβανομένων των nova-compute, nova-api, nova-volume, nova-network, Glance, Swift, Keystone και Horizon. Περιέχει δύο κάρτες διασύνδεσης δικτύου (NIC).

Base OS

Εγκατάσταση 64 bit έκδοση του Ubuntu 12.04 server διατηρώντας τις ακόλουθες ρυθμίσεις στο μυαλό .

- Δημιουργία του πρώτου χρήστη με το όνομα « localadmin » .
- Η Εγκατάσταση μας επιτρέπει να ρυθμίσουμε τη διεύθυνση IP για την πρώτο interface eth0 . Ρυθμίζουμε τις λεπτομέρειες της διεύθυνσης IP .
- Κατά την εγκατάσταση , επιλέγουμε μόνο το OpenSSH -server στο μενού πακέτα .

Επίσης, θα τρέχει το nova-volume σε αυτόν τον server και είναι ιδανικό για να έχουν ένα ειδικό partition για τη χρήση του nova-volume. Θα πρέπει να κάνουμε χειροκίνητα τη διαμέριση του δίσκου κατά την εγκατάσταση του Ubuntu Server και να δημιουργήσει ένα ειδικό partition με επαρκή χώρο για το σκοπό αυτό . Έχουμε αναφερθεί σε αυτό το διαμέρισμα στο υπόλοιπο του κεφαλαίου ως /dev/sda6. Επίσης, θα χρειαστεί να βεβαιωθούμε ότι ο τύπος της κατάτμησης έχει οριστεί ως Linux LVM (8e) χρησιμοποιώντας το fdisk, είτε κατά τη εγκατάσταση ή αμέσως μετά την εγκατάσταση έχει τελειώσει. Αν, επίσης, σκοπεύουμε να χρησιμοποιήσουμε ένα ειδικό partition, ως Swift backend, δημιουργούμε ένα άλλο διαμέρισμα για τον σκοπό αυτό και ακολουθούμε τις οδηγίες στην ενότητα "Swift Installation" παρακάτω.

Ενημερώνουμε το μηχάνημα χρησιμοποιώντας τις ακόλουθες εντολές.

```
sudo apt-get update
```

```
sudo apt -get upgrade
```

Εγκατάσταση bridge-utils:

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

Ρύθμιση του Δικτύου

Επεξεργασία του αρχείου /etc/network/interfaces, ώστε να μοιάζει με το παρακάτω:

```
auto lo
```

```
iface lo inet loopback
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 10.10.10.2
netmask 255.255.255.0
broadcast 10.10.10.255
gateway 10.10.10.1
dns-nameservers 10.10.8.3
auto eth1
iface eth1 inet static
address 192.168.3.1
netmask 255.255.255.0
network 192.168.3.0
broadcast 192.168.3.255
```

Κάνουμε επανεκκίνηση του

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

NTP Server

Εγκαθιστούμε το NTP πακέτο. Αυτός ο server θα λειτουργεί ως NTP server για όλα τα nodes. Η ώρα πρέπει να είναι συγχρονισμένη σε όλα τα OpenStack στοιχεία. Τρέχουμε τον NTP στον server 1 και τα υπόλοιπα nodes συγχρονίζονται πάνω του.

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

Στο αρχείο `/etc/ntp.conf` προσθέτουμε τις ακόλουθες γραμμές για το συγχρονισμού με εξωτερικό server. Αν πέσει η σύνδεση δικτύου, ο NTP server χρησιμοποιεί το δικό του hardware ρολόι ως κάλυψη.

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

Επανεκκινούμε τον NTP server

```
sudo service ntp restart
```

Οι servers θα αναγνωρίζονται είτε μέσω DNS είτε μέσω του `/etc/hosts` αρχείου.

Βάσεις δεδομένων

Μπορούμε να χρησιμοποιήσουμε MySQL, PostgreSQL είτε SQLite για Nova και Glance. Ανάλογα με το ποια θα εγκαταστήσουμε θα περάσουμε και τα αντίστοιχα πακέτα και θα ρυθμίσουμε τον database server.

MySQL

Εγκαθιστούμε mysql-server και python-mysqldb πακέτο

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

Δημιουργούμε root password για mysql. Το password που χρησιμοποιούμε εδώ είναι "pmygreatsecret"

Βάζουμε bind address από 127.0.0.1 σε 0.0.0.0 στο αρχείο /etc/mysql/my.cnf. Όπως φαίνεται παρακάτω :

```
bind-address = 0.0.0.0
```

Κάνουμε Restart τον MySQL server για να δούμε ότι ακούει σε όλα τα interfaces.

```
sudo restart mysql
```

Δημιουργία Βάσεων Δεδομένων

Δημιουργούμε MySQL βάσεις για να χρησιμοποιηθούν με τη nova, το glance και το keystone.

Δημιουργία της βάσης nova.

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

Δημιουργία του χρήστη novadbadmin.

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

Δικαιώματα στον novadbadmin για τη βάση "nova".

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

Δημιουργία password για το χρήστη "novadbadmin".

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

Δημιουργία της βάσης glance.

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

Δημιουργία του χρήστη glancedbadmin.

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

Παροχή πλήρων δικαιωμάτων στο glancedbadmin για τη βάση "glance".

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

Δημιουργία password για το χρήστη "glancedbadmin".

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

Δημιουργία της βάσης keystone.

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

Δημιουργία χρήστη keystoneadmin.

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

Παροχή πλήρων δικαιωμάτων στον keystoneadmin για τη βάση "keystone".

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

Δημιουργία password για το χρήστη "keystoneadmin".

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

Keystone

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

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

Άνοιγμα του αρχείου /etc/keystone/keystone.conf και αλλαγή της γραμμής

```
admin_token = ADMIN
```

έτσι ώστε να φαίνεται όπως παρακάτω:

```
admin_token = admin
```

Εφόσον η βάση θα αποθηκεύσει τις ρυθμίσεις keystone, αντικαθιστούμε την ακόλουθη γραμμή /etc/keystone/keystone.conf

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

με την παρακάτω

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

Επανεκκίνηση του Keystone:

```
sudo service keystone restart
```

Συγχρονίζουμε τη βάση τρέχοντας:

```
sudo keystone-manage db_sync
```

Κάνουμε Export τις μεταβλητές του συστήματος οι οποίες χρειάζονται ώστε να δουλέψουν με το OpenStack.

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

```
export SERVICE_TOKEN=admin
```

Μπορούμε φυσικά να τις ρίξουμε στο ~/.bashrc, ώστε να μη χρειάζεται να κάνουμε εξαγωγή κάθε φορά.

Δημιουργία Tenants

Αυτό επιτυγχάνεται εκτελώντας τα παρακάτω commands. Στην περίπτωσή μας tenants - admin and service.

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

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

Δημιουργία χρηστών

Δημιουργούμε τους χρήστες εκτελώντας τα παρακάτω commands. Εδώ θα φτιάξουμε τέσσερις χρήστες - admin, nova, glance και swift

```
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-ROLES

Αυτό επιτυγχάνεται με τα εξής commands : Εδώ έχουμε δύο ρόλους - admin and Member.

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

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

Εμφάνιση σε λίστα των Tenants, Users και Roles

Λίστα tenants

```
keystone tenant-list
```

```
keystone tenant-list
+-----+-----+-----+
|          id          |          name          | enabled |
+-----+-----+-----+
| 7f95ae9617cd496888bc412efdceabfd | admin                  | True   |
| c7970080576646c6959ee35970cf3199 | service                | True   |
+-----+-----+-----+
```

Λίστα χρηστών

keystone user-list

```
keystone user-list
```

id	enabled	email	name
1b986cca67e242f38cd6aa4bdec587ca	True	swift@foobar.com	swift
518b51ea133c4facadae42c328d6b77b	True	glance@foobar.com	glance
b3de3aeec2544f0f90b9cbfe8b8b7acd	True	admin@foobar.com	admin
ce8cd56ca8824f5d845ba6ed015e9494	True	nova@foobar.com	nova

Λίστα ρόλων

keystone role-list

```
keystone role-list
```

id	name
2bbe305ad531434991d4281aaaebb700	admin
d983800dd6d54ee3a1bleb9f2ae3291f	Member

Προσθήκη ρόλων σε Tenants

Αυτό επιτυγχάνεται με το εξής command:

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

Τα 'id' μπορεί να ληφθεί από το αποτέλεσμα των εντολών - keystone user-list, keystone tenant-list, keystone role-list.

Η προσθήκη ρόλου του ρόλου 'admin' στο χρήστη 'admin' στο tenant 'admin':

```
keystone user-role-add --user b3de3aeec2544f0f90b9cbfe8b8b7acd --role 2bbe305ad531434991d4281aaaebb700 --tenant_id 7f95ae9617cd496888bc412efdceabfd
```

Οι παρακάτω εντολές προσθέτουν το ρόλο 'admin' στους χρήστες 'nova', 'glance' και 'swift' για το tenant 'service'.

```
keystone user-role-add --user ce8cd56ca8824f5d845ba6ed015e9494 --role 2bbe305ad531434991d4281aaaebb700 --tenant_id c7970080576646c6959ee35970cf3199
```

```
keystone user-role-add --user 518b51ea133c4facadae42c328d6b77b --role 2bbe305ad531434991d4281aaaebb700 --tenant_id c7970080576646c6959ee35970cf3199
```

```
keystone user-role-add --user 1b986cca67e242f38cd6aa4bdec587ca -
-role 2be305ad531434991d4281aaaebb700 --tenant_id
c7970080576646c6959ee35970cf3199
```

Ο ρόλος 'Member' χρησιμοποιείται από το Horizon και το Swift. Άρα προστίθεται ανάλογα .

```
keystone user-role-add --user b3de3aeeec2544f0f90b9cbfe8b8b7acd --
roled983800dd6d54ee3a1b1eb9f2ae3291f --tenant_id
7f95ae9617cd496888bc412efdceabfd
```

βάζοντας ανάλογα το id ανά περίπτωση.

Δημιουργία Services

Τώρα θα γίνει δημιουργία των service με τα οποία θα πιστοποιούνται οι χρήστες. Τα nova-compute, nova-volume, glance, swift,keystone και ec2 είναι κάποια από τα services τα οποία θα δημιουργηθούν.

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

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

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

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

Όλα τα services έχουν ένα μοναδικό id το οποίο εμφανίζεται εκτελώντας

keystone service-list

```
keystone service-list
```

id	name	type	description
1e93ee6c70f8468c88a5cblb106753f3	nova	compute	OpenStack Compute Service
28fd92ffe3824004996a3e04e059d875	ec2	ec2	EC2 Service
7d4ec192dfal456996f0f4c47415c7a7	keystone	identity	OpenStack Identity Service
96f35e1112b143e59d5cd5d0e6a8b22d	swift	object-store	OpenStack Storage Service
f38f4564ff7b4e43a52b2f5c1b75e5fa	volume	volume	OpenStack Volume Service
fbafab6edcab467bb734380ce6be3561	glance	image	OpenStack Image Service

Το 'id' θα χρησιμοποιείται για να καθορίζει τα endpoints για το συγκεκριμένο service.

Δημιουργώντας Endpoints

Δημιουργία endpoints για κάθε ένα από τα services τα οποία δημιουργήθηκαν παραπάνω.

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

Για παράδειγμα για να δημιουργηθεί για το nova-compute, εκτελείται το παρακάτω command:

```
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'
```

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'
```

Glance

Εγκατάσταση Glance:

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

Ρύθμιση Glance

Το Glance χρησιμοποιεί SQLite “by default”. Η MySQL και η PostgreSQL μπορούν επίσης να χρησιμοποιηθούν με το Glance.

Επικολλούνται στο αρχείο `/etc/glance/glance-api-paste.ini` οι παρακάτω γραμμές:

```
admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%
```

όπου μπορούν να μπουν οι τιμές του χρηστή όπως φαίνεται παρακάτω:

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

Παρόμοιες αλλαγές και στο `/etc/glance/glance-registry-paste.ini`.


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

Για να ειπωθεί στο glance να χρησιμοποιεί το keystone για το authentication, προστίθενται οι ακόλουθες γραμμές στο τέλος του αρχείου .

```
[paste_deploy]
```

```
flavor = keystone
```

Άνοιγμα το αρχείο /etc/glance/glance-api.conf και προσθήκη στο τέλος του:

```
[paste_deploy]
```

```
flavor = keystone
```

Δημιουργία glance schema στην MySQL βάση:

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

```
sudo glance-manage db_sync
```

Restart του glance-api και του glance-registry αφού ολοκληρωθούν οι αλλαγές.

```
sudo restart glance-api
```

```
sudo restart glance-registry
```

Export ή εισαγωγή στο ~/.bashrc, για να μη χρειάζεται να γίνονται συνέχεια export οι μεταβλητές.

```
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 αν το glance ρυθμίστηκε σωστά με εκτέλεση της εντολής.

```
glance index
```

από το οποίο δεν υπάρχει έξοδος, αλλά με εκτέλεση του echo \$? και έξοδο 0 σημαίνει ότι έχει ρυθμιστεί σωστά και συνδέεται με το Keystone.

Nova

Για 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
```

Nova Configuration

Επεξεργασία του αρχείου /etc/nova/nova.conf να περιέχει τα ακόλουθα:

```
--dhcpbridge_flagfile=/etc/nova/nova.conf  
--dhcpbridge=/usr/bin/nova-dhcpbridge  
--logdir=/var/log/nova  
--state_path=/var/lib/nova  
--lock_path=/run/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=10.10.10.2
# 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

```

Δημιουργία ενός Physical Volume και ενός Volume Group.

```

sudo pvcreate /dev/sda6
sudo vgcreate nova-volumes /dev/sda6

```

Αλλαγή της ιδιοκτησίας του /etc/nova καταλόγου και των δικαιωμάτων του /etc/nova/nova.conf:

```

sudo chown -R nova:nova /etc/nova
sudo chmod 644 /etc/nova/nova.conf

```

Στο τέλος /etc/nova/api-paste.ini προστίθεται:

```

admin_tenant_name = %SERVICE_TENANT_NAME%
admin_user = %SERVICE_USER%
admin_password = %SERVICE_PASSWORD%

```

και συμπλήρωση με τις μεταβλητές του χρηστή

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

Δημιουργία του nova schema στη MySQL βάση.

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

Παροχή ένα εύρος από IPs για ανάθεση στα instances.

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

Export των παρακάτω μεταβλητών.

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

Επανεκκίνηση όλων των nova services.

```
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;
```

Για την επαλήθευση του setup εκτελείται

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

```
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
```

Εάν έχουμε παρόμοια έξοδο στο τερματικό μας, η υπηρεσία έχει στηθεί με επιτυχία.

OpenStack Dashboard

Η εγκατάσταση πραγματοποιείται με το εξής command:

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

και γίνεται επανεκκίνηση του Apache

```
service apache2 restart
```

Άνοιγμα ενός φυλλομετρητή και εισαγωγή της IP του server1. Θα πρέπει να εμφανιστεί η διεπαφή για να γίνει εισαγωγή στο Horizon. Πραγματοποίηση Login με username 'admin' και password 'admin'. Από το dashboard γίνεται δημιουργία κλειδιών, security groups, νέων μηχανημάτων κ.α.

Swift

Swift Installation

Τα πιο σημαντικά components είναι ο proxy, το account, το container και οι object servers.

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

Άλλα components είναι το xfsprogs (για συμβατότητα με XFS filesystem), python.pastedeploy (για πρόσβαση σε keystone), curl (για δοκιμή του swift).

Sudo apt-get install xfsprogs curl python-pastedeploy

Swift Storage Backends

Υπάρχουν δύο τρόποι για να δημιουργηθεί το storage, ή με προσάρτηση νέου δίσκου ή με partition στον υπάρχοντα δίσκο δημιουργώντας ένα loopback αρχείο και χρησιμοποιώντας το ως το storage device.

Partition ως το storage device

Αν δεν υπάρχει, δημιουργία ενός partition (e.g. /dev/sdb3), το οποίο γίνεται format σε xfs filesystem με χρήση parted ή fdisk, ώστε να χρησιμοποιηθεί ως backend.

Δημιουργία του mount point στο /etc/fstab αρχείο.

Εκτέλεση:

```
sudo fdisk /dev/sdb (Αντικατάσταση του /dev/sdb με την σωστή συσκευή)
```

και ακολούθηση των οδηγιών με τις σωστές επιλογές.

```
sudo mkfs.xfs -i size=1024 /dev/sdb3
```

```
sudo tune2fs -l /dev/sdb3 |grep -i inode
```

Δημιουργία του καταλόγου /mnt/swift_backend that για να χρησιμοποιηθεί ως mount point στο partition που φτιάξαμε.

```
sudo mkdir /mnt/swift_backend
```

Επεξεργασία του αρχείου /etc/fstab και προσθήκη της γραμμής:

```
/dev/sdb3 /mnt/swift_backend xfs
noatime,nodiratime,nobarrier,logbufs=8 0 0
```

για να μην χρειάζεται να γίνεται mount διαρκώς μετά από κάθε restart.

Loopback Αρχείο ως storage device

Δημιουργία ενός αρχείου με μηδενικά ως loopback device για το Swift storage backend. Χρησιμοποιώντας την εντολή disk copy δημιουργία του αρχείου swift-disk δεσμεύοντας ένα εκατομμύριο 1KiB blocks (976.56 MiB) σε αυτό. Έτσι υπάρχει αρχείο 1GiB το οποίο μπορεί να μεγαλώσει και θα έχει υποχρεωτικά XFS filesystem.

```
sudo dd if=/dev/zero of=/srv/swift-disk bs=1024 count=0
seek=1000000
```

```
sudo mkfs.xfs -i size=1024 /srv/swift-disk
```

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

```
swift-disk1: SGI XFS filesystem data (blkisz 4096, inosz 1024, v2
dirs)
```

Δημιουργία του καταλόγου /mnt/swift_backend that για να χρησιμοποιηθεί ως mount point στο partition που έγινε πριν.

```
sudo mkdir /mnt/swift_backend
```

Επεξεργασία του αρχείο /etc/fstab και προσθήκη τη γραμμή:

```
/dev/sdb3 /mnt/swift_backend xfs
noatime,nodiratime,nobarrier,logbufs=8 0 0
```

για να μην χρειάζεται να γίνεται mount διαρκώς μετά από κάθε restart

Χρησιμοποίηση του backend

Πριν γίνει mount το backend που θα χρησιμοποιηθεί, γίνεται δημιουργία κάποιων nodes να παίξουν το ρόλο των storage devices και ορισμός της ιδιοκτησίας του στο το 'swift' χρήστη και ομάδα.

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

```
pushd /mnt/swift_backend
```

```
sudo mkdir node1 node2 node3 node4
```

```
popd
```

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

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

```
sudo mkdir -p /etc/swift/account-server /etc/swift/container-
server /etc/swift/object-server
```

```
sudo mkdir /srv/node1/device /srv/node2/device /srv/node3/device
/srv/node4/device/run/swift
```

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

Για να τρέχει κάθε φορά που ξεκινάει ο υπολογιστής προσθήκη στο αρχείο `/etc/rc.local` των ακόλουθων γραμμών :

```
mkdir /run/swift
```

```
chown swift.swift /run/swift
```

Ρύθμιση Rsync

Το Rsync είναι υπεύθυνο για τη συντήρηση των object αντιγράφων και γενικότερα για ποικίλες λειτουργίες του swift. . Ρυθμίζεται για όλα τα storage nodes.

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

Αλλαγή του αρχείου `/etc/rsyncd.conf` όπως 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
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 του rsync.

```
sudo service rsync restart
```

Ρύθμιση Swift Components

Η Python χρησιμοποιεί το paste.deploy για τη διαχείριση των ρυθμίσεων. Οι προεπιλεγμένες ρυθμίσεις υπάρχουν στο τμήμα [DEFAULT], και όλες οι επιλογές που είναι εκεί μπορούν να παρακαμφθούν σε οποιοδήποτε άλλο τομέα αλλά ΜΟΝΟ ΜΕ ΤΗΝ ΣΥΓΚΕΚΡΙΜΕΝΗ ΣΥΝΤΑΞΗ set option_name = value.

Ορίστε ένα παράδειγμα ρύθμισης του paste.deploy:

```

[DEFAULT]
name1 = globalvalue
name2 = globalvalue
name3 = globalvalue
set name4 = globalvalue
[pipeline:main]
pipeline = myapp
[app:myapp]

```

```

use = egg:mypkg#myapp
name2 = localvalue
set name3 = localvalue
set name5 = localvalue
name6 = localvalue

```

Δημιουργία και επεξεργασία του /etc/swift/swift.conf και προσθήκη των γραμμών:

```

[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

```

Χρειάζεται το random string εάν θέλουμε να επεκταθεί την εγκατάσταση και να προστεθούν άλλα nodes. Φυλάσσεται αυτός ο κωδικός.

Δημιουργία του random string με την εντολή:

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

Configure Swift Proxy Server

Ο proxy server δρα ως φύλακας για το swift. Αναλαμβάνει να πιστοποιήσει το χρήστη . Η πιστοποίηση βεβαιώνει ότι ο χρήστης είναι αυτός που λέει και ποιος έχει πρόσβαση και που.

Η εξουσιοδότηση δίνεται από το keystone. Δημιουργία και ρύθμιση του

/etc/swift/proxy-server.conf ως εξής:

```

[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

```

```
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
```

Configure Swift Account Server

Η default ρύθμιση για το swift είναι /etc/swift/account-server.conf.

```
[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]
```

Τα configuration files βρίσκονται κάτω από τον /etc/swift/account-server.conf. Εδώ μπορούν να υπάρχουν και άλλα τέτοια αρχεία που καθένα αντιστοιχεί σε ένα device κάτω από τον /srv.

Τα περιεχόμενα του /etc/swift/account-server/1.conf είναι:

```
[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]
```

Δημιουργία αντιγράφων για τις υπόλοιπες συσκευές αν υπάρχουν και αλλαγή του inode του καθενός .

Configure Swift Container Server

To default swift container server configuration είναι /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]
```

Τα configuration files για το Container server βρίσκονται /etc/swift/container-server.conf. Αντίστοιχα και εδώ δημιουργία αντιγράφων /srv. Για παράδειγμα ονομάζονται 1.conf, 2.conf κ.ο.κ. Εδώ είναι το περιεχόμενο του αρχείου /etc/swift/container-server/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]

Για τα άλλα devices, (/srv/node2, /srv/node3, /srv/node4), δημιουργία των 2.conf, 3.conf και 4.conf.

Configure Swift Object Server

To default swift object server configuration για το αρχείο /etc/swift/object-server.conf είναι:

```
[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]
```

Τα Object server configuration files βρίσκονται κάτω από το /etc/swift/object-server.conf. Θα υπάρχουν όσα αντίγραφα όσα και devices

Το αρχείο /etc/swift/object-server/1.conf θα έχει:

```
[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]
```

Ομοίως και για τις άλλες συσκευές

Configure Swift Rings

Το Ring είναι ένα σημαντικό component για το swift. Συντηρεί πληροφορίες σχετικά με τη θέση τους μέσα στα objects, τα αντίγραφα και τα devices. Τώρα θα χτιστούν τα αντίστοιχα αρχεία για το object service, container service και account service.

Πάμε στο /etc/swift directory για την εκτέλεση των commands.

```
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
```

Οι δείκτες είναι οι επιθυμητοί αριθμοί για τα partitions, τα αντίγραφα και το χρόνο σε ώρες να περιορίσουν τη μετακίνηση ενός partition για πάνω από μία φορά. Για περισσότερες πληροφορίες ανατρέξτε στη man page για το swift.

Τώρα θα προστεθούν ζώνες και θα γίνει εξισορρόπηση των rings. Η σύνταξη δίνεται παρακάτω:

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

Εκτέλεση των παρακάτω εντολών για προσθήκη ζωνών και εξισορρόπηση των ring.

```
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
```

Starting Swift services

Για εκκίνηση των start swift και του REST API, εκτέλεση αυτών των εντολών.

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

Testing Swift

Το Swift δοκιμάζεται είτε με εντολές είτε μέσω του dashboard web interface (Horizon).

Μην ξεχνάμε να η κυριότητα του /etc/swift directory να ανήκει στο swift.swift.

```
sudo chown -R swift.swift /etc/swift
swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U service:swift
-K swift stat
```

Η εκτέλεση της παραπάνω εντολής δίνει κάτι σαν αυτό:

```
StorageURL:
http://127.0.0.1:8080/v1/AUTH_c7970080576646c6959ee35970cf3199
Auth Token: ba9df200a92d4a5088dcd6b7dcc19c0d
Account: AUTH_c7970080576646c6959ee35970cf3199
Containers: 1
Objects: 1
Bytes: 77
Accept-Ranges: bytes
X-Trans-Id: tx11c64e218f984749bc3ec37ea46280ee
```


Server2

Ο server αυτός τρέχει μόνο το nova-compute service.

Λειτουργικό Σύστημα

Εγκαθιστούμε 64 bit έκδοση των Ubuntu server 12.04

Ρύθμιση του Δικτύου

Εγκατάσταση των bridge-utils:

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

Επεξεργασία του αρχείου /etc/network/interfaces ώστε να μοιάζει έτσι:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 10.10.10.3
netmask 255.255.255.0
broadcast 10.10.10.255
gateway 10.10.10.1
dns-nameservers 10.10.8.3
auto eth1
iface eth1 inet static
address 192.168.3.2
netmask 255.255.255.0
network 192.168.3.0
broadcast 192.168.3.255
```

Επανεκκίνηση του δικτύου .

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

NTP Client

Εγκατάσταση του NTP πακέτου.

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

Διαμόρφωση του αρχείου /etc/ntp.conf προσθέτοντας τη παρακάτω γραμμή για συγχρονισμό με τον server1 και επανεκκίνηση της υπηρεσίας NTP.

```
server 10.10.10.2
```

```
sudo service ntp restart
```

Nova Components (nova-compute μόνο)

Εγκατάσταση των nova-components και των dependencies.

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

Διαμόρφωση του /etc/nova/nova.conf όπως ακολουθεί. Το αρχείο είναι ίδιο με αυτό στο Server1 (/etc/nova/nova.conf)

```
--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/run/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=10.10.10.2
# 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

```

Επανεκκίνηση του nova-compute στον Server2.

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

Έλεγχος αν ο compute node (Server2) τρέχει:

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

Εάν έχουμε στο τερματικό κάτι σαν το παρακάτω τότε έχει γίνει σωστά την εγκατάσταση.

```

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
nova-compute server2    nova      enabled     :-)  2012-04-21 10:22:27

```

Client1

Βασικό ΛΣ

Εγκατάσταση 64-bit έκδοσης Ubuntu 12.04 Desktop

Ρύθμιση του Δικτύου

Διαμόρφωση του αρχείου /etc/network/interfaces ώστε να δείχνει έτσι :

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 10.10.10.4
netmask 255.255.255.0
broadcast 10.10.10.255
gateway 10.10.10.1
dns-nameservers 10.10.8.3
```

NTP Client

Εγκατάσταση NTP πακέτου.

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

Άνοιγμα του αρχείου /etc/ntp.conf, προσθήκη της ακόλουθης γραμμής για το συγχρονισμό με το server1 και επανεκκίνηση της υπηρεσίας NTP.

```
server 10.10.10.2
sudo service ntp restart
```

Client Tools

Όπως έχει προαναφερθεί , πρόκειται για τη desktop εγκατάσταση των Ubuntu 12.04 η οποία θα χρησιμοποιηθεί για εργασίες όπως το “bundling of images”. Επίσης θα χρησιμοποιείται για τη διαχείριση της cloud υποδομής μέσω nova, glance και swift command line tools.

Εγκατάσταση των απαραίτητων command line tools με την τρέχουσα εντολή:

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

Εγκατάσταση του qemu-kvm

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

Εξαγωγή των ακόλουθων μεταβλητών συστήματος και τοποθέτηση αυτών στο ~/.bashrc.

```
Export SERVICE_TOKEN=admin
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=admin
export OS_AUTH_URL="http://10.10.10.2:5000/v2.0/"
export SERVICE_ENDPOINT=http://10.10.10.2:35357/v2.0
```

Εκτέλεση nova και glance commands για να ελεγχθεί αν συνδέονται με το OpenStack .

```
nova list
```

```
glance index
```

```
nova list
```

ID	Name	Status	Networks
25ee9230-6bb5-4eca-8808-e6b4e0348362	myinstance	ACTIVE	private=192.168.4.35
c939cb2c-e662-46e5-bc31-453007442cf9	myinstance1	ACTIVE	private=192.168.4.36

```
glance index
```

ID	Name	Disk Format	Container Format	Size
65b9f8e1-cde8-40e7-93e3-0866becfb9d4	windows	qcow2	ovf	7580745728
f147e666-990c-47e2-9caa-a5a21470cc4e	debian	qcow2	ovf	932904960
f3a8e689-02ed-460f-a587-dc868576228f	opensuse	qcow2	ovf	1072300032
aa362fd9-7c28-480b-845c-85a5c38ccd86	centoscli	qcow2	ovf	1611530240
49f0ec2b-26dd-4644-adcc-2ce047e281c5	ubuntuimage	qcow2	ovf	1471807488

OpenStack Dashboard

Άνοιγμα ενός φυλλομετρητή και πληκτρολόγηση πχ, <http://10.10.10.2>. Θα πρέπει να εμφανιστεί η οθόνη σύνδεσης . Είσοδος με τα στοιχεία username - admin και password - admin

Διαχείριση Image

Εισαγωγή

Υπάρχουν αρκετά έτοιμα images για το OpenStack διαθέσιμα από ποικίλες πηγές. Μπορούμε να κατεβάσουμε κάποια και να τα χρησιμοποιήσουμε για εξοικείωση με το OpenStack.

Για ένα επαγγελματική υλοποίηση θα χρειαστεί να χρησιμοποιήσουμε custom images, με custom σετ από εφαρμογές η ρυθμίσεις. Στο κεφάλαιο αυτό θα δείξουμε τη διαδικασία για τη δημιουργία των προσαρμοσμένων Linux images από δημοφιλείς διανομές από το μηδέν. Θα δούμε και μια απόπειρα για δημιουργία Windows images.

Ανάλογα με τη διανομή υπάρχουν μικρές διαφοροποιήσεις. Στα Ubuntu είναι ευκολότερο διότι παρέχεται το cloud-init πακέτο, το οποίο προσέχει τη ρύθμιση του instance την ώρα που φτιάχνεται. Το cloud-init χειρίζεται τα ssh κλειδιά για εισαγωγή χωρίς κωδικό login, να θέσουμε το host name κτλ. Το instance αποκτά τις εξειδικευμένες ρυθμίσεις από το Nova-compute συνδέοντας με ένα meta data interface το οποίο τρέχει στην IP 169.254.169.254.

Κατά τη διάρκεια της δημιουργίας ενός image από ένα distro που δεν έχει cloud-init ή κάτι αντίστοιχο, θα χρειαστεί να προσέχουμε για την εισαγωγή κλειδιών κα. Τρέχοντας ένα σετ εντολών κατά τη διάρκεια του boot από το rc.local.

Σε όλες τις περιπτώσεις, υποθέτουμε ότι έχουμε ένα KVM εγκατεστημένο για τη δημιουργία των images. Δουλεύουμε πάνω στο μηχάνημα 'client1' όπως έχουμε προαναφέρει.

Παρακάτω περιγράφεται η δημιουργία disk-images που αντιπροσωπεύουν ένα δίσκο χωρίς partitions.

Δημιουργία ενός Linux Image

Το πρώτο βήμα είναι η διασφάλιση επαρκή χώρου για την δημιουργία του στον Client1.

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

Εγκατάσταση ΛΣ

Κατέβασμα του iso με χρήση 'wget' ή από το browser.

Εκκίνηση ενός KVM instance με το ISO σε εικονικό CD-ROM. Αυτό ξεκινάει την διαδικασία εγκατάστασης και παρακάτω θέτουμε ένα VNC στην πόρτα 0.

```
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 σύνδεση στο VM μέσω VNC (αριθμός εμφάνισης :0) και τέλος εγκατάστασης.

Για παράδειγμα, με IP 10.10.10.4 του client1 δίνουμε:

```
vncviewer 10.10.10.4 :0
```

Κατά τη δημιουργία των Linux images, φτιάχνουμε ένα ext4 partition το οποίο το προσαρτούμε στο swap partition. Μετά το τέλος της εγκατάστασης γίνεται επανέναρξη του VM εκτελώντας την παρακάτω εντολή.

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

Σε αυτό το σημείο γίνονται ότι αλλαγές θέλουμε.

Ubuntu

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install openssh-server cloud-init
```

Αφαιρούμε διαγράφουμε το network persistence rules από το /etc/udev/rules.d γιατί θα παρουσιάζεται ως διαφορετική από την eth0 στις ρυθμίσεις.

```
sudo rm -rf /etc/udev/rules.d/70-persistent-net.rules
```

Fedora

```
yum update
yum install openssh-server
chkconfig sshd on
```

Επεξεργαζόμαστε το αρχείο /etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE="eth0"
BOOTPROTO=dhcp
NM_CONTROLLED="yes"
ONBOOT="yes"
```

Αφαιρούμε διαγράφουμε το network persistence rules από το /etc/udev/rules.d γιατί θα παρουσιάζεται ως διαφορετική από την eth0 στις ρυθμίσεις.

```
sudo rm -rf /etc/udev/rules.d/70-persistent-net.rules
```

Τερματίζουμε το virtual machine.

Εφόσον το Fedora δεν έχει το cloud-init ή παρεμφερές, θα πρέπει να κάνουμε αυτά τα έξτρα βήματα

στο /etc/rc.local προσθέτουμε τις ακόλουθες γραμμές πριν τη γραμμή "touch /var/lock/subsys/local"

```
depmod -a
modprobe acpihp
# simple attempt to get the user ssh key using the meta-data
service
mkdir -p /root/.ssh
echo >> /root/.ssh/authorized_keys
```

```

curl -m 10 -s http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key| grep 'ssh-rsa' >> /root/.ssh/authorized_keys
echo "AUTHORIZED_KEYS:"
echo "*****"
cat /root/.ssh/authorized_keys
echo "*****"

```

CentOS 6 and RHEL 6

Εγκατάσταση του SSH server, Curl και όσων πακέτων χρειάζονται. Προσθήκη των παρακάτω γραμμών στο /etc/rc.local.

```

echo >> /root/.ssh/authorized_keys
curl -m 10 -s http://169.254.169.254/latest/meta-data/public-
keys/0/openssh-key | grep 'ssh -rsa' >>
/root/.ssh/authorized_keys
echo "AUTHORIZED_KEYS:"
echo "*****"
cat /root/.ssh/authorized_keys
echo "*****"

```

```

#τιιάχνουμε το /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO=dhcp
NM_CONTROLLED="yes"
ONBOOT="yes"

```

Διαγράφη του αρχείου που βρίσκεται στο /etc/udev/rules.d

```
rm -rf /etc/udev/rules.d/70-persistent-net.rules
```

Uploading the Linux image

Ανέβασμα της εικόνας image

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

Δημιουργία ενός Windows Image

Για τη δημιουργία image στον Client1 εξασφαλίζουμε επαρκή χώρο

```
kvm-img create -f qcow2 windowsserver.img 20G
```


Εγκατάσταση ΛΣ

Το OpenStack για το δίσκο χρησιμοποιεί virtio interface. Ως εκ τούτου, το λειτουργικό σύστημα πρέπει να έχει οδηγούς για virtio. Από προεπιλογή, ο Windows Server 2008 ISO δεν διαθέτει τα προγράμματα οδήγησης για virtio. Κατεβάστε την εικόνα iso που περιέχει virtio οδηγούς από την ακόλουθη θέση <http://alt.fedoraproject.org/pub/alt/virtio-win/latest/images/bin> και να το επισυνάψουμε κατά την εγκατάσταση

Ξεκινάμε την εγκατάσταση εκτελώντας:

```
sudo kvm -m 1024 -cdrom windows2008.iso -drive  
file=windowsserver1.img,if=virtio -boot d -drive file=virtio-  
win-0.1-22.iso,index=3,media=cdrom -device virtio-net-pci -net  
nic -net user -nographic -vnc :5
```

Όταν ολοκληρωθεί η εγκατάσταση ζητά να επιλέξουμε μια συσκευή σκληρού δίσκου και δεν θα δούμε καμία διαθέσιμη συσκευή. Κάνουμε κλικ στο "Load drivers" κάτω αριστερά και φορτώνουμε τα προγράμματα οδήγησης από την περιήγηση στο δευτερεύον CDROM στο οποίο ο δίσκος του οδηγού virtio είναι φορτωμένος.

Αφού ολοκληρωθεί η εγκατάσταση, εκκινούμε άμεσα και εγκαθιστούμε τυχόν πρόσθετες εφαρμογές που χρειάζεται να εγκαταστήσουμε και κάνουμε αλλαγές διαμόρφωσης που πρέπει να κάνουμε. Επίσης, βεβαιωνόμαστε ότι το RDP έχει ενεργοποιηθεί, διότι θα είναι ο μόνος τρόπος που μπορούμε να συνδεθούμε με Windows. Το τείχος προστασίας των Windows θα πρέπει να ρυθμιστεί ώστε να επιτρέπει τις εισερχόμενες ICMP και RDP συνδέσεις.

Uploading the Windows image

Διακόπτουμε τη λειτουργία του VM και ανεβάζουμε την εικόνα στο OpenStack

```
glance add name="windows" is_public=true container_format=ovf  
disk_format=qcow2 < windowsserver.img
```

Διαχείριση Instance

Εισαγωγή

Ένα instance είναι μια εικονική μηχανή η οποία τροφοδοτείται από τον OpenStack σε έναν από τους διακομιστές nova-compute. Όταν ξεκινάμε ένα instance, μια σειρά από δράσεις που ενεργοποιούνται στα διάφορα components του OpenStack. Κατά τη διάρκεια των κύκλων ζωής ενός instance, κινείται μέσα σε διάφορα στάδια, όπως φαίνεται στο παρακάτω διάγραμμα: Οι ακόλουθες διασυνδέσεις μπορεί να χρησιμοποιηθούν για τη διαχείριση των instances στην nova.

- εντολές Nova
- Προσαρμοσμένες εφαρμογές που έχουν αναπτυχθεί με τη χρήση Nova APIs
- Προσαρμοσμένες εφαρμογές που έχουν αναπτυχθεί με τη χρήση APIs EC2

Openstack Command Line Tools

Η Nova έχει μια δέσμη εργαλείων της γραμμής εντολών για τη διαχείριση της εγκατάστασης OpenStack. Οι εντολές αυτές θα σας βοηθήσουν να διαχειριστούμε τις εικόνες, τα instances, την αποθήκευση, τη δικτύωση κλπ. Μερικές εντολές που σχετίζονται με τη διαχείριση των περιπτώσεων που δίνονται παρακάτω.

Creation of Key Pairs

Οι OpenStack υπηρεσίες πιστοποιούνται και εγκρίνονται μέσω keystone server. Το Keystone παρέχει ένα διακριτικό και μια υπηρεσία καταλόγου που περιέχει πληροφορίες σχετικά με τις παραμέτρους των υπηρεσιών στις οποίες ένας χρήστης είναι εξουσιοδοτημένος. Κάθε χρήστης έχει ένα διακριτικό token και service catalog που δημιουργήθηκε για αυτό. Αυτό μπορείτε να το κατεβάσετε από το OpenStack Dashboard. Θα πρέπει, επίσης, να δημιουργήσουμε ένα keypair που αποτελείται από ιδιωτικό κλειδί / δημόσιο κλειδί για να είμαστε σε θέση να ξεκινήσουμε τα instances του OpenStack. Με αυτά τα κλειδιά συνδεόμαστε χωρίς κωδικούς πρόσβασης. Τα keypairs μπορούν επίσης να παραχθούν χρησιμοποιώντας τις ακόλουθες εντολές.

```
ssh-keygen
```

```
cd .ssh
```

```
nova keypair-add --pub_key id_rsa.pub mykey
```

Οι παραπάνω δημιουργούν ένα keypair με όνομα mykey. Το ιδιωτικό κλειδί id_rsa σώζεται τοπικά στον ~/.ssh. Μπορούμε να δούμε τα διαθέσιμα keypairs με την εντολή nova keypair-list.

nova keypair-list

```
nova keypair-list
+-----+-----+
| Name | Fingerprint |
+-----+-----+
| mykey | b0:18:32:fa:4e:d4:3c:1b:c4:6c:dd:cb:53:29:13:82 |
| mykey2 | b0:18:32:fa:4e:d4:3c:1b:c4:6c:dd:cb:53:29:13:82 |
+-----+-----+
```

Επίσης, κατά την εκτέλεση του «ssh-keygen» μπορούμε να καθορίσουμε μια προσαρμοσμένη θέση και προσαρμοσμένα ονόματα αρχείων για τα keypairs που θέλουμε να δημιουργήσουμε.

Για να διαγράψετε μια υπάρχουσα keypair:

nova keypair-delete mykey2

Εκκίνηση και Διαχείριση instances

Υπάρχουν πολλές εντολές που βοηθούν στη διαχείριση των instances. Εδώ είναι μερικά παραδείγματα:

```
$ nova boot --flavor 1 --image 9bab7ce7-7523-4d37-831f-c18fbc5cb543 --key_name mykey myinstance
```

```
$ nova boot --flavor 1 --image 9bab7ce7-7523-4d37-831f-c18fbc5cb543 --key_name mykey myinstance ←
```

Property	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	instance-00000002
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
accessIPv4	
accessIPv6	
adminPass	FaUPM6EEBT8F
config_drive	

```
$ nova list
```

ID	Name	Status	Networks
25ee9230-6bb5-4eca-8808-e6b4e0348362	myinstance	ACTIVE	private=192.168.4.35
c939cb2c-e662-46e5-bc31-453007442cf9	myinstance1	ACTIVE	private=192.168.4.36

```
$ nova delete 25ee9230-6bb5-4eca-8808-e6b4e0348362
```

```
$ nova list
```

ID	Name	Status	Networks
c939cb2c-e662-46e5-bc31-453007442cf9	myinstance1	ACTIVE	private=192.168.4.34

```
$ nova console-log myinstance
```

Για σύνδεση χωρίς κωδικό με ssh στο instance:

```
ssh -i <private_key> username@<ip_address>
```

Ο Τύπος του VM έχει συνέπειες για το μέγεθος του σκληρού δίσκου, την ποσότητα της μνήμης RAM και τον αριθμό των επεξεργαστών που διατίθενται για τα instance. Ελέγχουμε τους τύπους των διαθέσιμων VM .

```
nova flavor-list
```

Νέα flavours δημιουργούνται με την παρακάτω εντολή.

```
sudo nova-manage flavor create <args> [options]
```

και διαγράφονται με τον εξής τρόπο

```
sudo nova-manage flavor delete <args> [options]
```

OpenStack Dashboard (Horizon)

Χρησιμοποιώντας το OpenStack Dashboard, μπορεί κανείς να διαχειριστεί διάφορες υπηρεσίες OpenStack. Μπορεί να χρησιμοποιηθεί για τη διαχείριση των instances και εικόνων τη δημιουργία keypairs, διαχείριση Swift containers κλπ. Το OpenStack Dashboard είναι προσβάσιμο μέσω

```
http://<ip_address>
```

Login

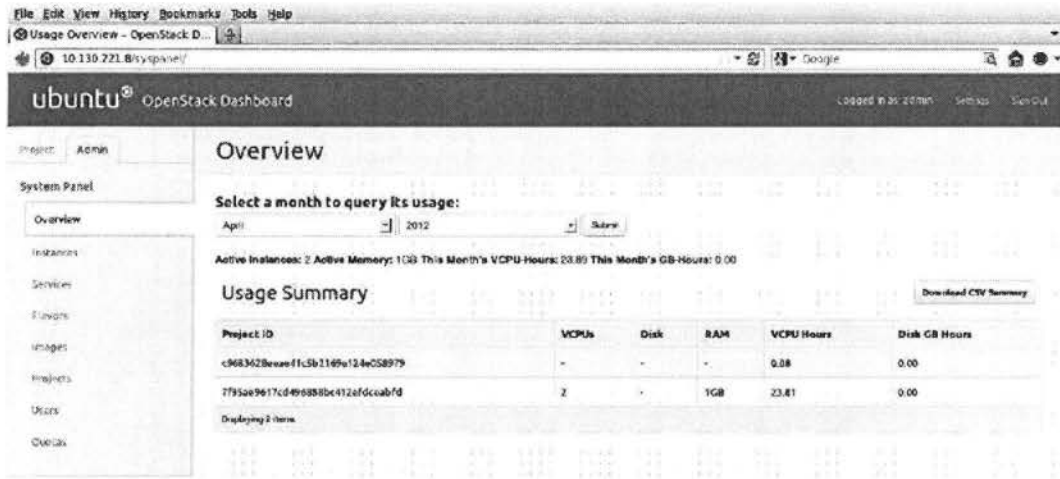
Εισαγωγή στο dashboard με username "admin" και password "admin".



Εικ 5 Σελίδα Login

Επισκόπηση χρήστη

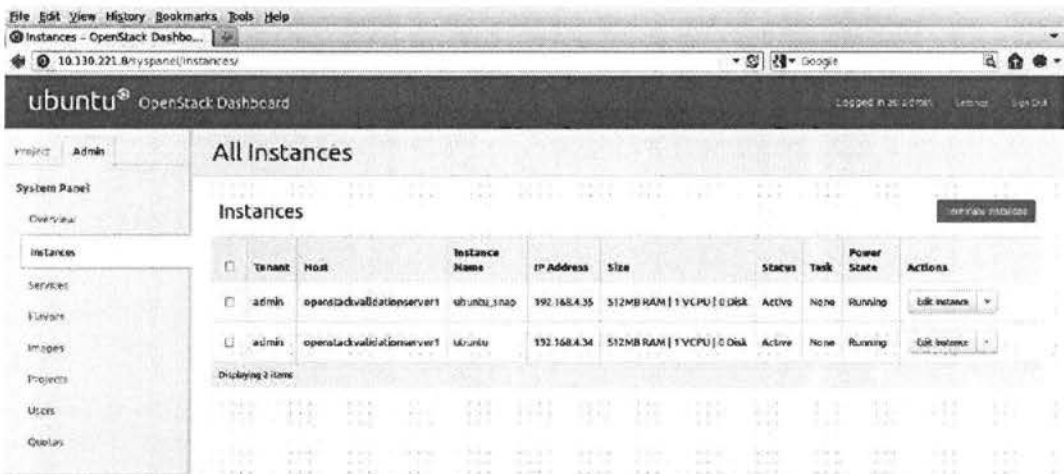
Μετά την καταγραφή, ανάλογα με τα δικαιώματα πρόσβασης, ο χρήστης έχει τη δυνατότητα πρόσβασης σε συγκεκριμένα έργα. Το παρακάτω είναι μια σελίδα επισκόπησης για ένα έργο που ανήκει στον 'admin' χρήστη.



Εικ 6 Σελίδα Επισκόπησης

Instances

Η σελίδα βάζει σε διάταξη τα instances που εκτελούνται αυτή τη στιγμή και ανήκουν στον χρήστη "admin". Από τη σελίδα αυτή, μπορεί κανείς να τερματίσει, να διακόψει, κάνει επανεκκίνηση να συνδεθεί με VNC κονσόλα κλπ.



Εικ 7 Σελίδα Στιγμιότυπων

Services

Εδώ βλέπουμε τη λίστα με τις υπηρεσίες.

The screenshot shows the OpenStack Dashboard interface. The browser address bar displays '10.130.221.8/syspanel/services/'. The page title is 'Services - OpenStack Dashboard'. The main content area is titled 'Services' and contains a table with the following data:

Name	Service	Host	Enabled
nova	compute	10.130.221.8	Enabled
glance	image	10.130.221.8	Enabled
volume	volume	10.130.221.8	Enabled
ec2	ec2	10.130.221.8	Enabled
swift	objectstore	10.130.221.8	Enabled
keystone	identity (native backend)	10.130.221.8	Enabled

At the bottom of the table, it says 'Displaying 6 items'.

Εικ 8 Σελίδα Υπηρεσιών

Flavors

Αυτή η σελίδα απεικονίζει τα διαθέσιμα flavors που μπορούν να χρησιμοποιηθούν για να ξεκινήσει ένα instance. Κάποιος μπορεί επίσης να δημιουργήσει προσαρμοσμένα flavors σε αυτή τη σελίδα.

The screenshot shows the OpenStack Dashboard interface for the 'Flavors' page. The browser address bar displays '10.130.221.8/syspanel/flavors/'. The page title is 'Flavors - OpenStack Dashboard'. The main content area is titled 'Flavors' and contains a table with the following data:

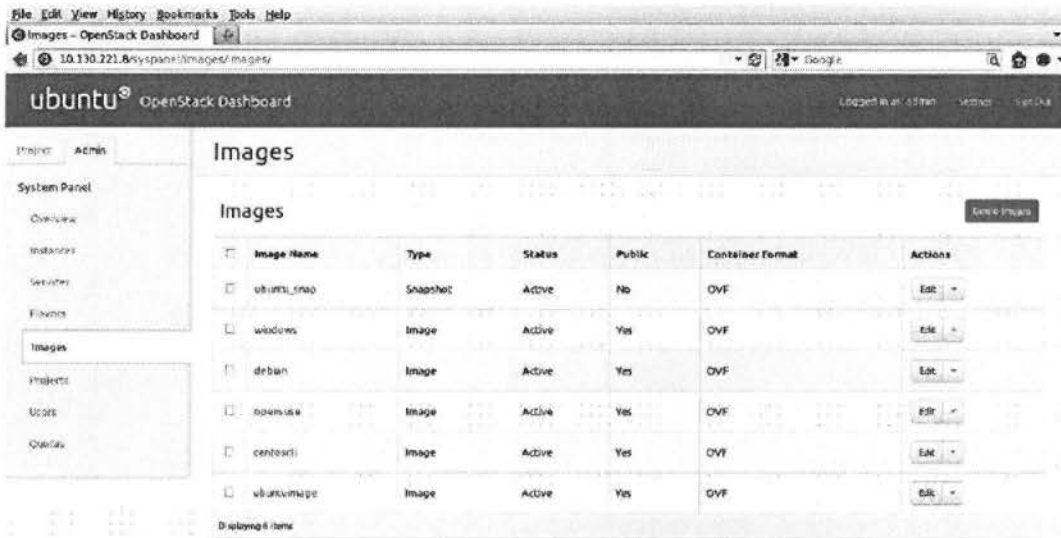
ID	Flavor Name	VCPUs	Memory	Root Disk	Ephemeral Disk	Actions
5	m1.large	8	16384	10	160	Create Flavor
4	m1.large	4	8192	10	80	Create Flavor
3	m1.medium	2	4096	10	40	Create Flavor
2	m1.small	1	2048	10	20	Create Flavor
1	m1.tiny	1	512	-	-	Create Flavor

At the bottom of the table, it says 'Displaying 5 items'.

Εικ 9 Σελίδα Flavors

Images

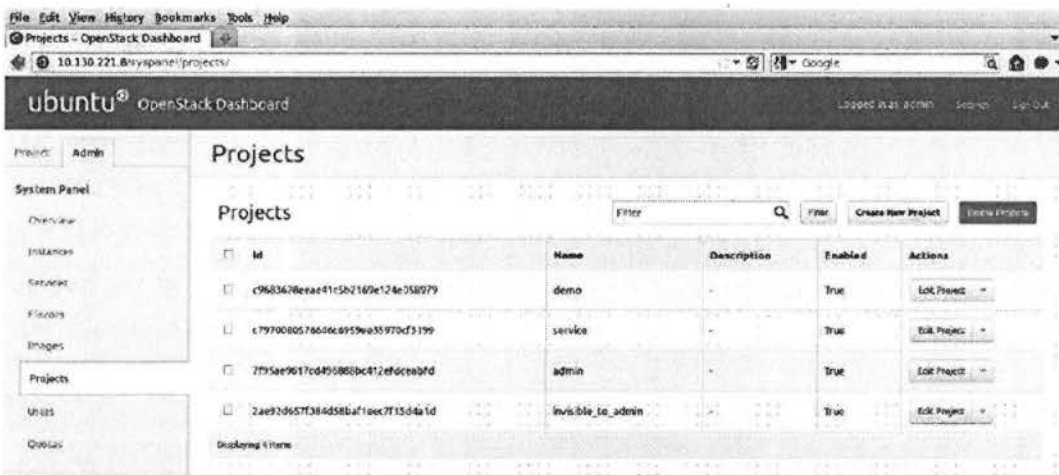
Αυτή η σελίδα απαριθμεί τις διαθέσιμες εικόνες για το «διαχειριστή» των χρηστών. Κάποιος μπορεί επίσης να διαγράψει τις εικόνες, εφόσον δεν απαιτούνται.



Εικ 10 Σελίδα Images

Projects

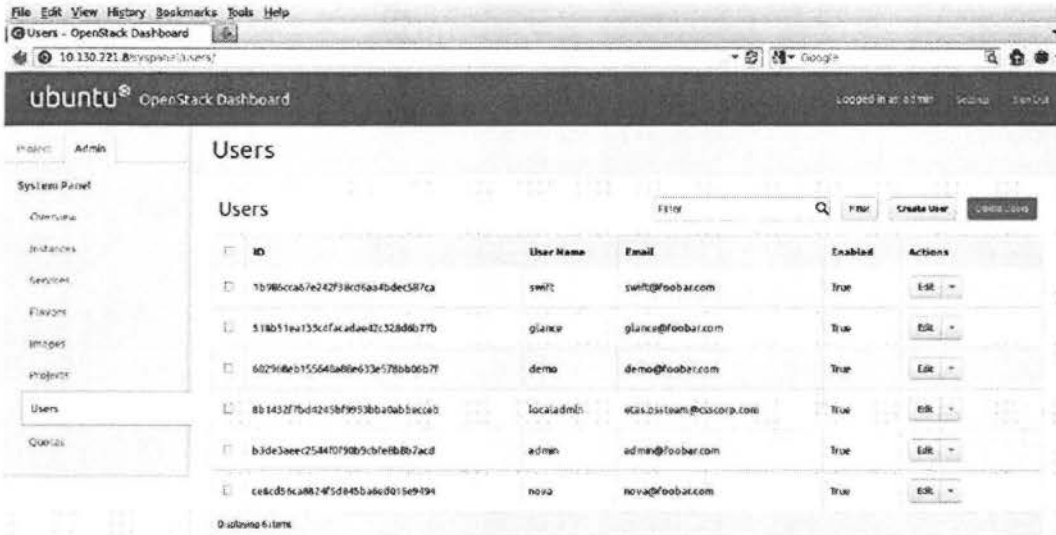
Αυτή η σελίδα απαριθμεί τα διαθέσιμα έργα (tenants-ενοικιαστές) που έχουν δημιουργηθεί. Κάποιος μπορεί επίσης να δημιουργήσουν νέα έργα, να εκχωρήσετε χρήστες στα έργα κλπ..



Εικ 11 Σελίδα Projects

Users

Η σελίδα αυτή απαριθμεί τους χρήστες που έχουν δημιουργηθεί. Δίνει τη δυνατότητα για διαγραφή δημιουργία απενεργοποίηση κα.

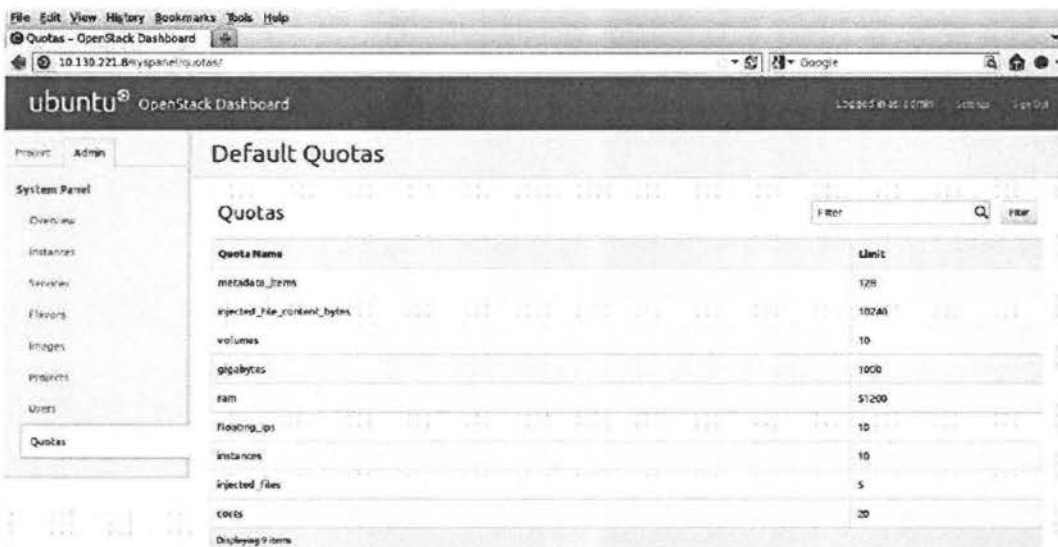


ID	User Name	Email	Enabled	Actions
1b9b6cra07e242f38d6aa4bdec587ca	swift	swift@foobar.com	True	Edit
518b51ea155c4facd4e42c528d6077b	glance	glance@foobar.com	True	Edit
6029d8eb15564a8be633e578bb0657f	demo	demo@foobar.com	True	Edit
8b14327bd4245bf993bbabab5ccab0	localadmin	localadmin@foobar.com	True	Edit
b3de3aee25440f90b9c4e1b8b7acd	admin	admin@foobar.com	True	Edit
ce8cd5fca812475d845b46c0015e9194	nova	nova@foobar.com	True	Edit

Εικ 12 Σελίδα Χρηστών

Quotas

Η σελίδα αυτή δείχνει τον αριθμό των πόρων που κατανέμονται ανά χρήστη. CPU, μνήμη, αποθηκευτικός χώρος κ.α.



Quota Name	Limit
metadata_items	128
injected_file_content_bytes	10240
volumes	10
gigabytes	1000
ram	51200
floating_ips	10
instances	10
injected_files	5
cores	20

Εικ 13 Σελίδα Αριθμού διαθέσιμων πόρων

Επισκόπηση Project

Δίνει μια συνολική εικόνα για το project 'admin'. Μπορούν επίσης να ελεγχθούν μετρήσεις των πόρων του συστήματος.

The screenshot shows the 'Instance Overview' page for the 'admin' project. It includes a sidebar with navigation options like 'Overview', 'Instances & Volumes', 'Images & Snapshots', 'Access & Security', 'Object Store', and 'Containers'. The main content area displays usage statistics for April 2012: 1 Active Instance, 512MB Active Memory, 23.15 This Month's VCPU-Hours, and 0.00 This Month's GB-Hours. Below this is a 'Usage Summary' table with the following data:

Instance Name	VCPU's	Disk	RAM	Uptime
ubuntu	1	-	512MB	17 hours, 50 minutes

Εικ 14 Επισκόπηση Project

Instances & Volumes

Αυτή η σελίδα απαρτιθμεί όλα τα instances που ανήκουν σε διάφορους χρήστες του έργου, instance ιδιότητες κλπ. παρατίθενται επίσης όλοι οι τόμοι που έχουν δημιουργηθεί και η κατάστασή τους. Κάποιος μπορεί επίσης να δημιουργήσει νέα volumes και τα επισυνάπτει στα instances σε αυτή τη σελίδα.

The screenshot shows the 'Instances & Volumes' page. The 'Instances' section contains the following table:

Instance Name	IP Address	Size	Status	Task	Power State	Actions
ubuntu_1nvp	192.168.4.35	512MB RAM 1 VCPU 0 Disk	Active	None	Running	Edit instance
ubuntu	192.168.4.54	512MB RAM 1 VCPU 0 Disk	Active	None	Running	Edit instance

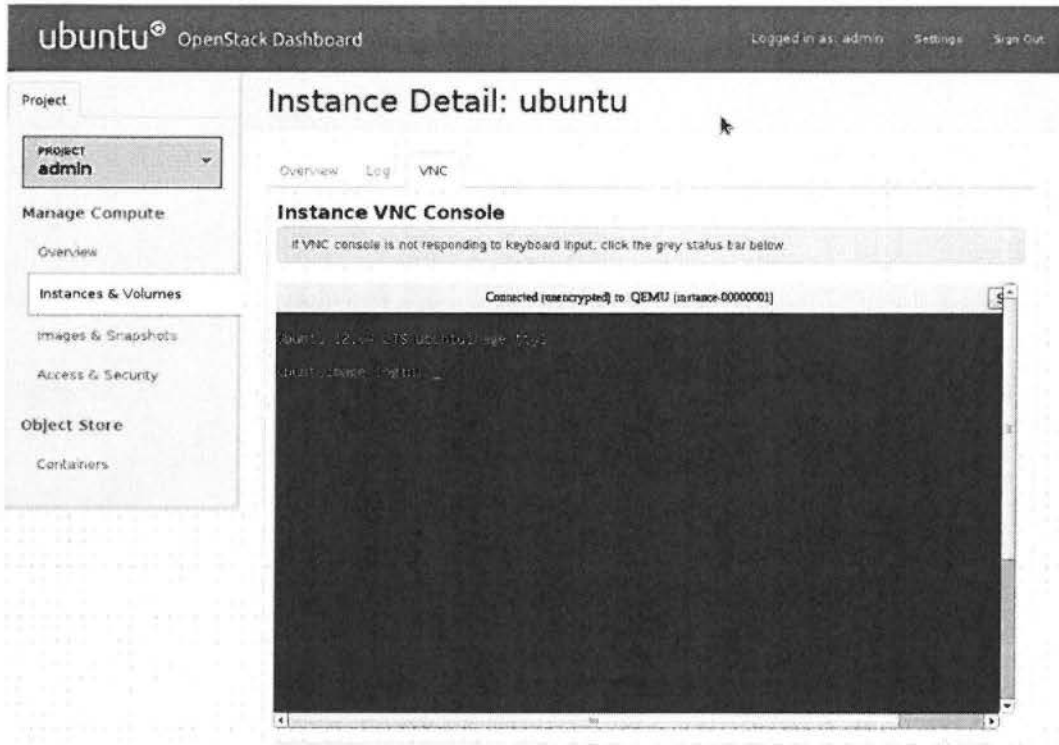
The 'Volumes' section shows a table with one volume:

Name	Destination	Size	Status	Attachments	Actions
New Volume	-	2 GB	In-use	Instance 7db4c0d4-776f-42e1-9f38-e59c3a318276 (ovirt.kit)	Edit Attachments

Εικ 15 Σελίδα Στιγμιότυπων και Τόμων

Instances - VNC Console

Δίνει τη δυνατότητα της σύνδεσης μέσω κονσόλας από τον περιηγητή μέσω VNC.



Εικ 16 Σελίδα Κονσόλας VNC Στιγμιότυπου

Images & Snapshots

Αυτή η σελίδα απαριθμεί τα προσαρμοσμένα images που έχουν φορτωθεί. Κάποιος μπορεί να επεξεργαστεί τις ιδιότητες των images, να διαγράψει και να ξεκινήσει νέα instances αυτών. Αυτή η σελίδα απαριθμεί επίσης τα στιγμιότυπα που έχουν ληφθεί από instances και volumes.

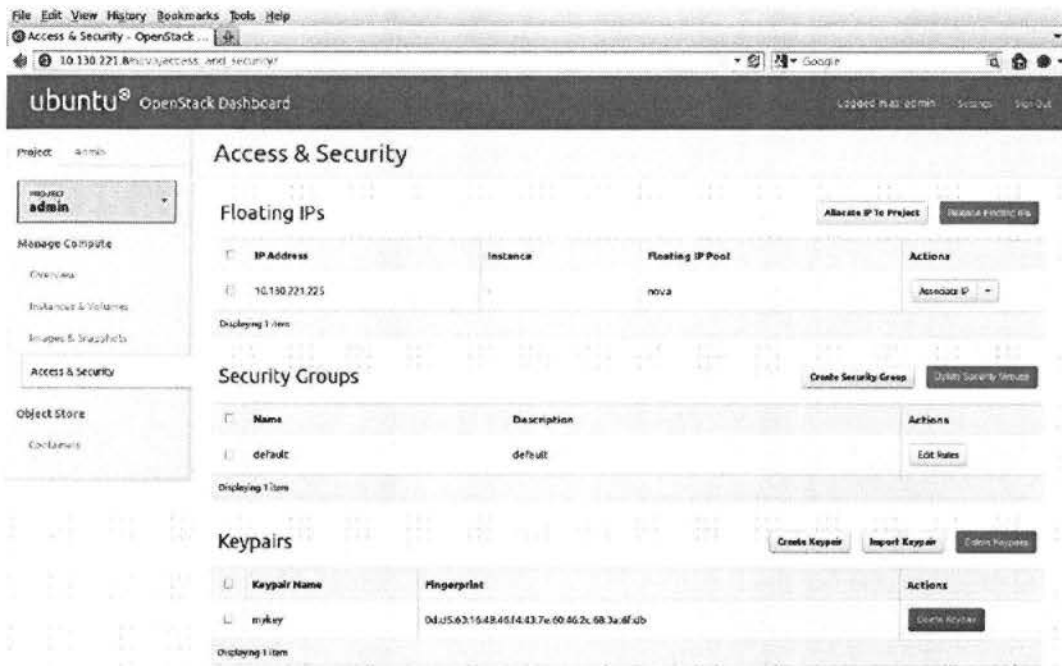
The screenshot shows the 'Images & Snapshots' page in the Ubuntu OpenStack Dashboard. The page is organized into three main sections:

- Images:** A table listing five images: 'windows', 'debian', 'openstack', 'centoscli', and 'ubuntuimage'. Each row includes columns for Image Name, Type, Status, Public, Container Format, and Actions (with a 'Launch' button).
- Instance Snapshots:** A table listing one snapshot: 'ubuntu_snap'. It includes columns for Image Name, Type, Status, Public, Container Format, and Actions (with a 'Launch' button).
- Volume Snapshots:** A section indicating 'No items to display'.

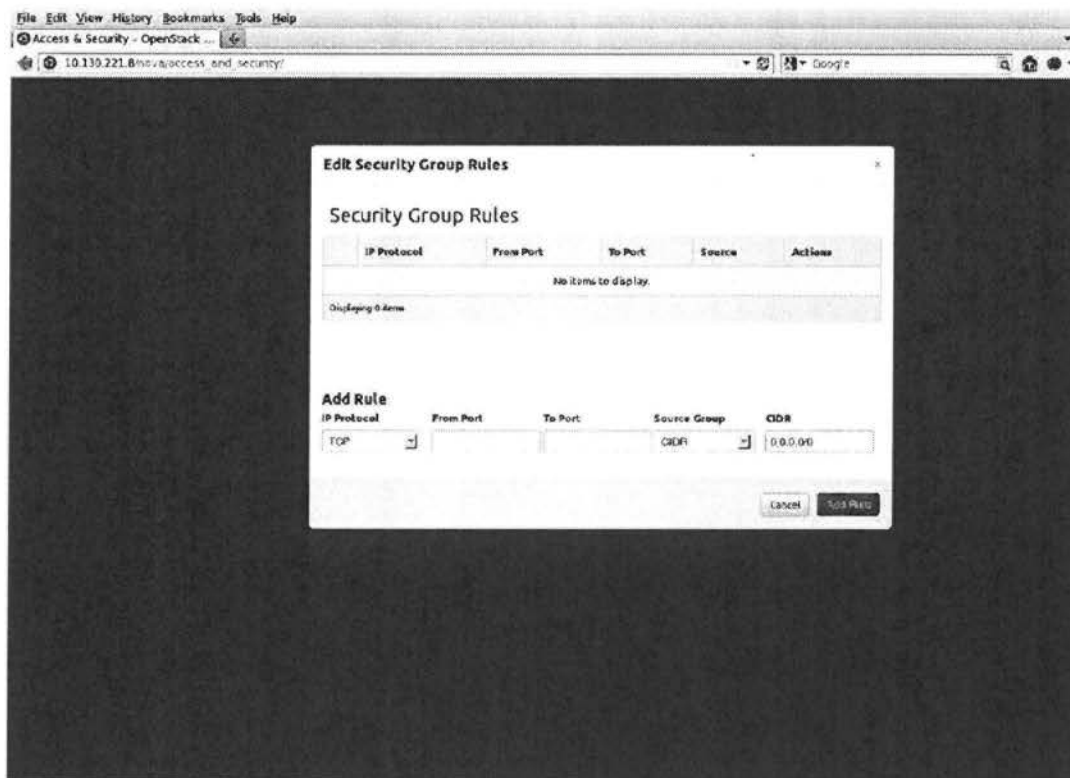
Εικ 17 Σελίδα Images και Snapshots

Access & Security

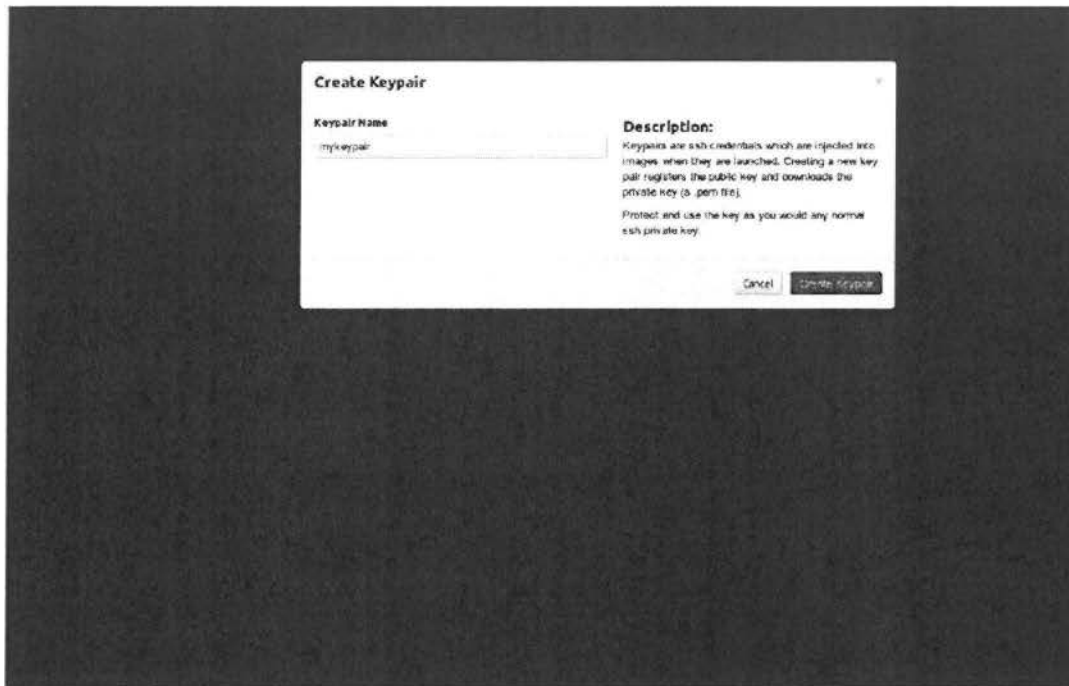
Σε αυτή τη σελίδα, μπορεί κανείς να δεσμεύσει και να αναθέσει floating διευθύνσεις IP, στα instances. Νέες ομάδες ασφαλείας μπορούν να δημιουργηθούν καθώς και να τροποποιηθούν οι κανόνες που ανήκουν σε κάθε ομάδα ασφαλείας.



Εικ 18 Σελίδα Πρόσβασης και Ασφάλειας



Εικ 19 Σελίδα δημιουργίας Κανόνων Ασφαλείας



Εικ 20 Σελίδα δημιουργίας Keypair

Containers & Objects

Σε αυτή τη σελίδα, μπορεί κανείς να δημιουργήσει ή διαγράψει containers, απαριθμήσει τα αντικείμενα, πραγματοποίηση αποστολή ή λήψη αντικειμένων, καθώς και διαγραφή αντικειμένων.



Εικ 21 Σελίδα ελέγχου Containers



Εικ 22 Σελίδα ελέγχου Objects

Διαχείριση Storage

Nova-volume

Αυτή η σελίδα απαριθμεί τις προσαρμοσμένες εικόνες που έχουν φορτωθεί. Κάποιος μπορεί να επεξεργαστεί τις ιδιότητες της εικόνας, να διαγράψει και να ξεκινήσει νέα instances των εικόνων. Αυτή η σελίδα απαριθμεί επίσης τα στιγμιότυπα που έχουν ληφθεί από τα instances και τα volumes.

Το Nova-volume παρέχει μόνιμη αποθήκευση. Η αποθήκευση των instances είναι προσωρινή και έτσι τα δεδομένα που παράγονται και αποθηκεύονται στο δίσκο τους χάνονται με τον τερματισμό τους. Για αυτό το λόγο χρησιμοποιείται το Nova-volume ώστε τα δεδομένα να παραμένουν και μετά τον τερματισμό.

Αλληλεπίδραση με τον ελεγκτή αποθήκευσης

Θα πρέπει να διασφαλιστεί ότι θα τρέξει το αρχείο novarc με την εντολή source πριν την εκτέλεση οποιασδήποτε από τις ακόλουθες εντολές. Οι εν λόγω εντολές αναφέρονται σε μία ζώνη που ονομάζεται «nova», η οποία δημιουργήθηκε στο κεφάλαιο «Εγκατάσταση και Διαμόρφωση». Το έργο είναι το «proj», όπως αναφέρεται στα άλλα κεφάλαια.

Δημιουργία ενός volume 10 GB

```
nova volume-create --display_name myvolume 10
```

Εμφάνιση μιας λίστας των volumes

```
nova volume-list
```

Η έξοδος της:

ID	Status	Display Name	Size	Volume Type	Attached to
1	in-use	New Volume	20	None	7db4cb64-7f8f-42e3-9f58-e59c9a31827d
4	available	volume1	10	None	
5	available	myvolume	10	None	
6	available	myvolume1	10	None	

Προσάρτηση ενός volume με ένα ενεργό instance

```
nova volume-attach 857d70e4-35d5-4bf6-97ed-bf4e9a4dcf5a <volume-id> /dev/vdb
```

Ένα volume μπορεί να προσαρτηθεί μόνο σε ένα instance κάθε φορά. Όταν το nova volume-list δείχνει την κατάσταση του volume ως «διαθέσιμο», αυτό σημαίνει ότι δεν είναι προσαρτημένο σε κάποιο instance και έτοιμο να χρησιμοποιηθεί. Αν εκτελεστεί η εντολή nova volume-list μπορούμε να δούμε ότι η κατάσταση αλλάζει από "διαθέσιμο" σε "σε χρήση" εάν προσαρτηθεί σε ένα instance επιτυχώς. Όταν ένα volume είναι προσαρτημένο σε ένα instance εμφανίζεται ως ένας πρόσθετος δίσκος στο vm-instance. Μπορούμε να συνδεθούμε και να κάνουμε mount το δίσκο, να κάνουμε διαμόρφωση και να το χρησιμοποιήσουμε.

Αποπροσάρτηση ενός volume από ένα instance.

```
nova volume-detach 857d70e4-35d5-4bf6-97ed-bf4e9a4dcf5a <volume-id>
```

Τα δεδομένα στα volume παραμένουν ακόμα κι αν αποπροσαρτηθεί από το instance ή εκείνο διαγραφεί. Έτσι μπορούμε να το προσαρτήσουμε σε ένα άλλο μηχάνημα.

Παρότι δηλώθηκε το /dev/vdb ως συσκευή στο instance, μπορεί να διαφέρει γιατί το ΛΣ δίνει ένα δικό του όνομα στη συσκευή. Το όνομα μπορεί να βρεθεί κοιτάζοντας στα device nodes στο /dev ή στο syslog την ώρα που γίνεται η προσάρτηση.

Swift

Το Swift είναι μια αξιόπιστη, κατανεμημένη, μαζικά επεκτεινόμενη άμορφη υπηρεσία που μπορεί να χρησιμοποιηθεί για την αποθήκευση και αρχειοθέτηση των αντικειμένων. Το Swift παρέχει ένα REST interface. Μπορούν επίσης να χρησιμοποιηθούν οι εντολές κονσόλας του Swift ως διεπαφή για το OpenStack object store service.

Για πληροφορίες σχετικά με το swift account, container and objects εκτελείται:

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U service:swiftuser -K swiftpasswd stat
Account: AUTH_43b42dae-dc0b-4a4b-ac55-97de614d6e6e
Containers: 1
Objects: 1
Bytes: 1124
Accept-Ranges: bytes
X-Trans-Id: txb21186a9eef64ed295ale95896a0fc72
```

Για πληροφορίες σχετικά με ένα συγκεκριμένο container (mycontainer):

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd stat mycontainer
```

Για πληροφορίες σχετικά με ένα object (abc123.txt) μέσα σε container (mycontainer):

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd stat mycontainer abc123.txt
```

Λίστα διαθέσιμων containers στο account:

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd list
```

Λίστα με όλα τα containers των οποίων τα ονόματα ξεκινάνε από 'my':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd -- - prefix=my list
```

Λίστα με objects μέσα σε container 'mycontainer':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd -- - prefix=my list mycontainer
```

Ανέβασμα των αρχείων 'abc.txt' και 'xyz.txt' στο 'mycontainer':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd upload - mycontainer
/path/abc.txt /path/xyz.txt
```

Κατέβασμα όλων των objects από όλα τα containers:

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd -all download
```

Κατέβασμα των objects από το container 'mycontainer':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd download mycontainer
```

Κατέβασμα του 'abc.txt' και του 'xyz.txt' από το container 'mycontainer':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U
service:swiftuser -K swiftpasswd download mycontainer abc.txt
xyz.txt
```


Διαγραφή όλων των objects σε όλα τα containers:

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U  
service:swiftuser -K swiftpasswd -all delete
```

Διαγραφή όλων των objects στο container 'mycontainer':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U  
service:swiftuser -K swiftpasswd delete --mycontainer
```

Διαγραφή των αρχείων 'abc.txt' και 'xyz.txt' από το container 'mycontainer':

```
$ swift -v -V 2.0 -A http://127.0.0.1:5000/v2.0/ -U  
service:swiftuser -K swiftpasswd delete --mycontainer abc.txt  
xyz.txt
```

Διαχείριση Δικτύου

Εισαγωγή

Στο OpenStack, η δικτύωση ελέγχεται από ένα στοιχείο που ονομάζεται «nova-network». Αυτό αλληλοεπιδρά με το nova-compute για να εξασφαλίσει ότι τα instances έχουν τις σωστές ρυθμίσεις δικτύου για να επικοινωνούν μεταξύ τους καθώς και με τον εξωτερικό κόσμο. Τα OpenStack instances μπορούν να έχουν 2 τύπους διευθύνσεων IP που συσχετίζονται με αυτά: Ιδιωτική διεύθυνση IP (στατική) και τη διεύθυνση IP Δημόσια (floating). Οι ιδιωτικές διευθύνσεις IP που χρησιμοποιούνται συνήθως για την επικοινωνία μεταξύ των vm (εσωτερική) και οι δημόσιες διευθύνσεις IP χρησιμοποιούνται για την επικοινωνία τους με τον έξω κόσμο (εξωτερική ή Internet). Οι λεγόμενες δημόσιες διευθύνσεις IP δεν χρειάζεται κατά ανάγκη να είναι route-able στο Internet, μπορούν ακόμη και να ανήκουν σε ένα εταιρικό δίκτυο LAN. Οι ρυθμίσεις του δικτύου στο εσωτερικό των instances γίνεται με ιδιωτικές IP διευθύνσεις.

Η συσχέτισμός μεταξύ της ιδιωτικής IP και της δημόσιας IP και η απαραίτητη δρομολόγηση χειρίζονται από το nova-network -δίκτυο και τα instances δεν χρειάζεται να τις γνωρίζουν .

Η υπηρεσία nova-network παρέχει 3 διαφορετικές επιλογές για την διαχείριση δικτύου. Αυτή τη στιγμή μπορεί να χρησιμοποιηθεί, μόνο μια από αυτές τις επιλογές.

- Flat Network
- Flat DHCP Network
- VLAN Network

Το VLAN Network είναι το πιο πλούσιο σε χαρακτηριστικά και ιδανικό για υλοποιήσεις παραγωγής (production deployment), ενώ οι άλλες δύο επιλογές χρησιμοποιούνται για εξοικείωση με το OpenStack και αν δεν είναι διαθέσιμα τα VLAN switches για σύνδεση των διάφορων στοιχείων της υποδομής του OpenStack .

Ο τύπος του δικτύου ρυθμίζεται μέσα από τις επιλογές του αρχείου nova.conf . Εάν δεν οριστεί κάποιος network manager τότε χρησιμοποιείται ο προεπιλεγμένος, που είναι ο VLANManager.

```
--network_manager = nova.network.manager.FlatManager
--network_manager = nova.network.manager.FlatDHCPManager
--network_manager = nova.network.manager.VlanManager
```

Σε κάθε μια από αυτές τις περιπτώσεις, εκτελούνται οι παρακάτω εντολές για τον ορισμό ιδιωτικών και δημοσίων διευθύνσεων IP για χρήση από τα instances:

```
sudo nova-manage network create private --
fixed_range_v4=192.168.4.3/27 --num_networks=1 --bridge=br100 --
bridge_interface=eth1 --network_size=32

sudo nova-manage floating create --ip_range=10.10.10.224/27
```

Η public IP την οποία πρόκειται να ανατεθεί σε ένα instance πρώτα πρέπει να δεσμευτεί και αυτό το πετυχαίνεται με αυτήν την εντολή:

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

που δεσμεύει την 10.10.10.225

Η ανάθεση πετυχαίνεται με την εκτέλεση της εντολής:

```
nova add-floating-ip <instance-name> 10.10.2.225
```

Ασφάλεια

Επισκόπηση Ασφαλείας

Το OpenStack παρέχει φίλτρα για τα vm μέσω των ομάδων ασφαλείας(security-groups) και δημιουργεί τα κατάλληλα iptables rules. Μια ομάδα ασφαλείας είναι ένα επώνυμο σύνολο κανόνων που ισχύουν για τα εισερχόμενα πακέτα για τα instances. Μπορείτε να ορίσετε μια ομάδα ασφαλείας, κατά την έναρξη ενός instance.

Κάθε ομάδα ασφαλείας μπορεί να έχει πολλαπλούς κανόνες

να σχετίζεται με αυτήν. Κάθε κανόνας ορίζει την source IP / δικτύου, τον τύπο του πρωτοκόλλου, τις destination ports κλπ. Κάθε πακέτο που ταιριάζει σε αυτές τις παραμέτρους επιτρέπεται να εισέλθει. Τα υπόλοιπα πακέτα αποκλείονται.

Μια ομάδα ασφαλείας που δεν έχει κανένα κανόνα προκαλεί μπλοκάρισμα όλης της εισερχόμενης κίνησης. Ο μηχανισμός μόνο παρέχει φιλτράρισμα εισόδου και δεν παρέχει κανένα φιλτράρισμα εξερχόμενης κίνησης. Ως αποτέλεσμα, όλες οι εξερχόμενες κινήσεις επιτρέπονται. Αν χρειαστεί να εφαρμοστεί φιλτράρισμα εξερχόμενης κίνησης, θα πρέπει να εφαρμόσουμε ότι και μέσα στο παράδειγμα (κατά τη διαδικασία ομαδοποίησης) χρησιμοποιώντας ένα firewall. Το OpenStack dashboard μας επιτρέπει να διαχειριστούμε ομάδες ασφαλείας, αλλά και μας επιτρέπει να καθορίσουμε μια ομάδα ασφαλείας, κατά την έναρξη ενός instance.

Επιπλέον επιτυγχάνεται με τη χρήση των commands όπως το `nova secgroup-add-rule` etc. Παρακάτω δίνονται nova commands για τη διαχείριση των security groups.

#Δημιουργία ενός security group με όνομα "myservers".

```
nova secgroup-create <name> <description>
```

```
nova secgroup-create myservers my-default-server-group
```

Προσθήκη ενός κανόνα-rule στο security group "myservers" να επιτρέπει icmp και tcp πακέτα από την 192.168.1.1.

```
nova secgroup-add-rule myservers tcp 22 22 192.168.1.1/0
```

```
nova secgroup-add-rule myservers icmp -1 -1 192.168.1.1/0
```

Για ένα Windows instance, προσθέτουμε ένα rule να δέχεται τις εισερχόμενες RDP αιτήσεις

```
nova secgroup-add-rule myservers tcp 3389 3389 192.168.1.1/0
```

Βλέπουμε το σύνολό των κανόνων με την εντολή.

```
$ nova secgroup-list-rules myservers
```

Αφαιρούμε τον κανόνα για ssh πακέτα από την source ip 192.168.1.1 από το security group "myservers"

```
nova secgroup-delete-rule myservers ssh 22 22 192.168.1.1
```

Διαγραφή του security group "myservers"

```
nova secgroup-delete myservers
```

Ξεκινώντας ένα instance που ανήκει στο security group "myservers".

```
nova boot --flavor 1 --image 9bab7ce7-7523-4d37-831f-c18fbc5cb543 --key_name mykey myinstance --security_groups myservers
```

Εάν δεν αναθέσουμε κάποιο security group, το instance ανήκει σε ένα προ-φτιαγμένο security group το "default". Τους κανόνες και εδώ μπορούμε να τους διαμορφώσουμε στα μέτρα μας.

OpenStack Commands

Nova Commands

Το nova είναι το command line interface για το OpenStack Compute API.

Usage: nova command [options] [args]

Commands:

help	Display help about this program or one of its subcommands.
actions	Retrieve server actions.
backup-schedule	Show or edit the backup schedule for a server.
backup-schedule-delete	Delete the backup schedule for a server.
boot	Boot a new server.
delete	Immediately shut down and delete a server.
diagnostics	Retrieve server diagnostics.
flavor-list	Print a list of available 'flavors' (sizes of servers).
image-create	Create a new image by taking a snapshot of a running server.
image-delete	Delete an image.
image-list	Print a list of available images to boot from.
ip-share	Share an IP address from the given IP group onto a server.
ip-unshare	Stop sharing an given address with a server.
ipgroup-create	Create a new IP group.
ipgroup-delete	Delete an IP group.
ipgroup-list	Show IP groups.
ipgroup-show	Show details about a particular IP group.
list	List active servers.
pause	Pause a server.
reboot	Reboot a server.
rebuild	Shutdown, re-image, and re-boot a server.
rename	Rename a server.
rescue	Rescue a server.
resize	Resize a server.
resize-confirm	Confirm a previous resize.
resize-revert	Revert a previous resize (and return to the previous VM).
resume	Resume a server.
root-password	Change the root password for a server.
show	Show details about the given server.
suspend	Suspend a server.
unpause	Unpause a server.
unrescue	Unrescue a server.
zone	Show or edit a child zone.
zone-add	Add a new child zone.
zone-delete	Delete a zone.
zone-info	Get this zones name and capabilities.
zone-list	List the children of a zone.

Glance Commands

To Glance εἶναι to command line interface γὰρ to OpenStack Imaging service.

Usage: glance command [options] [args]

Commands:

help command	Output help for one of the commands below
add	Adds a new image to Glance
update	Updates an image's metadata in Glance
delete	Deletes an image from Glance
index	Return brief information about images in Glance
details	Return detailed information about images in Glance
show	Show detailed information about an image in Glance
clear	Removes all images and metadata from Glance

Member Commands:

image-members	List members an image is shared with
member-images	List images shared with a member
member-add	Grants a member access to an image
member-delete	Revokes a member's access to an image
members-replace	Replaces all membership for an image

Swift Commands

Swift is the command line interface for OpenStack Object Store service.

```
Usage: swift command [options] [args]
```

Commands:

```
stat      Displays information for the account, container, or object depending
          on the args given (if any).
list      Lists the containers for the account or the objects for a container.
upload    Uploads to the given container the files and directories specified by
          the remaining args.
post      Updates meta information for the account, container, or object
          depending on the args given.
download  Downloads everything in the account (with --all), or everything in
          a container, or a list of objects depending on the args given.
delete    Deletes everything in the account (with --all), or everything in
          a container, or a list of objects depending on the args given.
```

Keystone Commands

To Keystone είναι το command line interface για το OpenStack Identity service.

```
Usage: keystone command [options] [args]
```

Commands:

```
catalog      List service catalog, possibly filtered by service.
ec2-credentials-create  Create EC2-compatible credentials for user per tenant
ec2-credentials-delete  Delete EC2-compatible credentials
ec2-credentials-get     Display EC2-compatible credentials
ec2-credentials-list    List EC2-compatible credentials for a user
endpoint-create        Create a new endpoint associated with a service
endpoint-delete        Delete a service endpoint
endpoint-get           Find endpoint filtered by a specific attribute or service type
endpoint-list          List configured service endpoints
role-create           Create new role.
role-delete           Delete role
role-get              Display role details
role-list             List all roles, or only those granted to a user.
service-create        Add service to Service Catalog
service-delete        Delete service from Service Catalog
service-get           Display service from Service Catalog
service-list          List all services in Service Catalog
tenant-create         Create new tenant
tenant-delete         Delete tenant
tenant-get            Display tenant details
tenant-list           List all tenants
tenant-update         Update tenant name, description, enabled status
token-get             Display the current user token
user-create           Create new user
user-delete           Delete user
user-get              Display user details.
user-list             List users
user-password-update  Update user password
user-role-add         Add role to user
user-role-remove      Remove role from user
user-update           Update user's name, email, and enabled status
discover              Discover Keystone servers and show authentication protocols and
help                  Display help about this program or one of its subcommands.
```

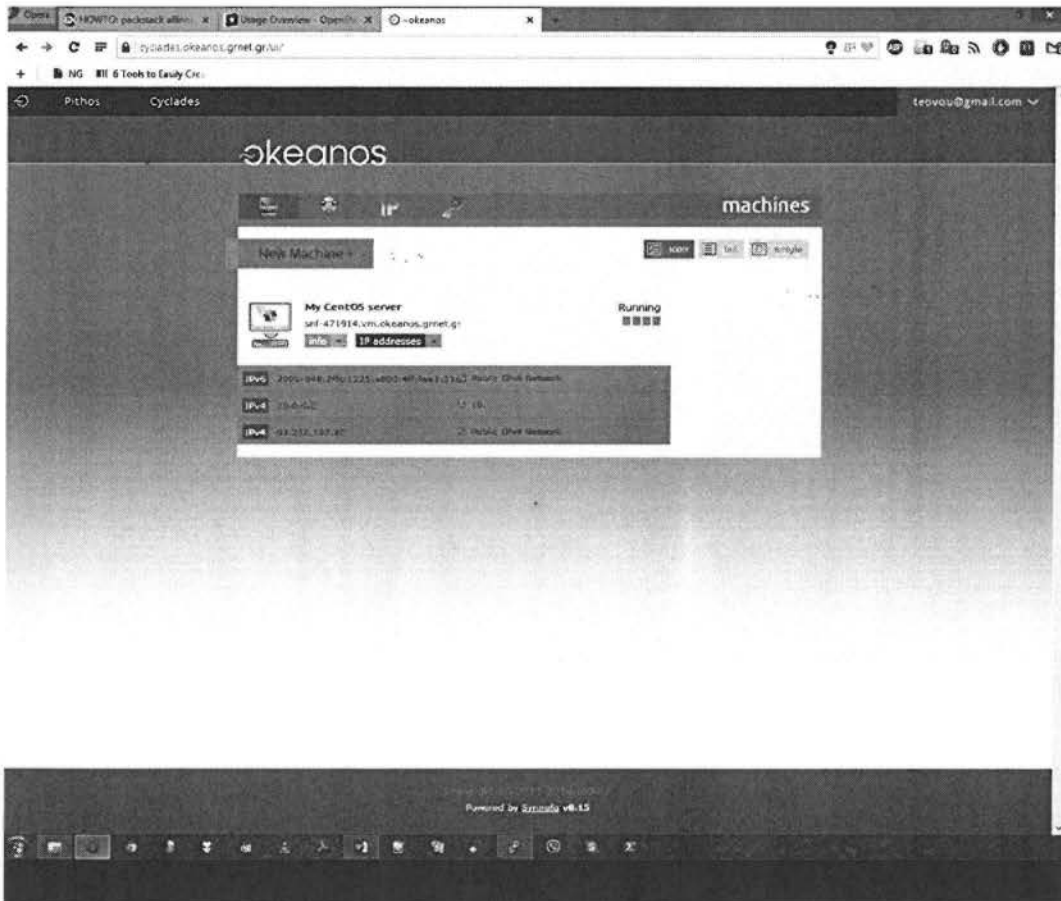
[22]

Υλοποίηση

Το στήσιμο της υλοποίησης έγινε πάνω στον Ωκεανό που είναι και αυτό μια πλατφόρμα IaaS όπως και το Openstack. Ο λόγος που χρησιμοποιήθηκε ο Ωκεανός είναι η έλλειψη κατάλληλης υποδομής πάνω στην οποία θα μπορούσε να στηθεί το OpenStack.

Για το στήσιμο του OpenStack χρησιμοποιήθηκε το RDO, το οποίο δίνει την δυνατότητα να στηθεί το OpenStack με την χρήση απλών script τα οποία παρόλα αυτά είναι παραμετροποιήσιμα ώστε να μπορεί να γίνει προσαρμογή της εγκατάστασης σύμφωνα με τις εκάστοτε ανάγκες.

Για την υλοποίηση δημιουργήσαμε ένα εικονικό μηχάνημα στην πλατφόρμα του Ωκεανού. Για λειτουργικό επιλέξαμε το CentOS 6.5 x64. Το επιλεγμένο λειτουργικό πρέπει να είναι πάντα αρχιτεκτονικής x64, ώστε να μπορεί να στηθεί πάνω του το OpenStack.



Εικ 23 Centos Server

Το script που χρησιμοποιήθηκε κάνει χρήση του προϋπάρχοντος δικτύου

```
# packstack --allinone --provision-all-in-one-ovs-bridge=n
```

Θεωρούμε ότι ο Server έχει IP 192.168.0.25/24 και gateway 192.168.0.1. Αφού ολοκληρωθεί η εγκατάσταση του OpenStack γίνεται επεξεργασία του αρχείου /etc/sysconfig/network-scripts/ifcfg-br-ex ώστε να μοιάζει έτσι

```
DEVICE=br-ex  
  
DEVICETYPE=ovs  
  
TYPE=OVSBridge  
  
BOOTPROTO=static  
  
IPADDR=192.168.0.25 # Old eth0 IP since we want the network  
restart to not kill the connection, otherwise pick something  
outside your dhcp range  
  
NETMASK=255.255.255.0 # μάσκα δικτύου  
  
GATEWAY=192.168.0.1#  
  
DNS1=192.168.0.1 # nameserver  
  
ONBOOT=yes
```

Γίνεται αλλαγή του αρχείου /etc/sysconfig/network-scripts/ifcfg-eth0 ώστε να μοιάζει με το παρακάτω:

```
DEVICE=eth0  
  
HWADDR=52:54:00:92:05:AE # η MAC της κάρτας δικτύου  
  
TYPE=OVSPort  
  
DEVICETYPE=ovs  
  
OVS_BRIDGE=br-ex  
  
ONBOOT=yes
```

Γίνεται επανεκκίνηση της υπηρεσίας δικτύου

```
service network restart
```


Είναι απαραίτητο να γίνει επανεκκίνηση της υπηρεσίας δικτύου πριν γίνει η ρύθμιση του δρομολογητή γιατί καταστρέφει και ξαναδημιουργεί το `br-ex`, το οποίο προκαλεί την διαγραφή της διεπαφής του δρομολογητή στο `qrouter-* netns`, η οποία δεν θα ξαναδημιουργηθεί αν δεν ξαναγίνει ρύθμιση του `gateway`.

```
. keystoneadmin
quantum router-gateway-clear router1
quantum subnet-delete public_subnet
```

Πρέπει να δημιουργηθεί ένα νέο δημόσιο υποδίκτυο με εύρος δέσμευσης IP έξω από το εύρος του DHCP και να οριστεί την προεπιλεγμένη πύλη του εξωτερικού δικτύου.

```
quantum subnet-create --name public_subnet --enable_dhcp=False -
-allocation-pool=start=192.168.122.10,end=192.168.122.20 --
gateway=192.168.122.1 public 192.168.122.0/24
quantum router-gateway-set router1 public
```

[23]

Εκτίμηση Αποτελεσμάτων

Η εγκατάσταση μιας πλατφόρμας IaaS είναι ακόμα σε αρχικά στάδια και παρουσιάζει πολλές ιδιαιτερότητες. Ανάλογα με τους διαθέσιμους πόρους και τον σκοπό της υλοποίησης, η εγκατάσταση και η αρχιτεκτονική μπορεί να διαφέρει σημαντικά. Υπάρχουν αρκετές διαθέσιμες επιλογές στην αγορά, όπως το OpenStack, το Eucalyptus, το Synnefo και το Apache CloudStack.

Έγιναν προσπάθειες εγκατάστασης του Synnefo, η εγκατάσταση του οποίου παρουσίασε μια σειρά δυσκολιών, κυρίως λόγω της έλλειψης ικανοποιητικού οδηγού εγκατάστασης, καθώς και προβλημάτων λόγω ασυμβατότητας μεταξύ `components` και ελλείψεων πολλών `dependencies`.

Το Eucalyptus μας άφησε θετικές εντυπώσεις λόγω ευκολίας στην εγκατάσταση και όμορφου UI.

Το OpenStack που επιλέχθηκε τελικά για την υλοποίηση που πραγματοποιήθηκε είναι η πιο πλήρης πλατφόρμα και λόγω της εξαιρετικής υποστήριξης με βιβλία που αναφέρονται σε

αυτό, αναλυτικούς οδηγούς καθώς και forum από όλες τις εταιρείες που συμμετέχουν στην ανάπτυξή του, η εγκατάσταση του αν και πολύπλοκη καθίσταται, σχετικά απλή.

Βιβλιογραφία

[1] «<http://en.wikipedia.org>,» [Ηλεκτρονικό].

Available: <http://en.wikipedia.org/wiki/OpenStack>.

[2] «<http://www.openstack.org>,» [Ηλεκτρονικό].

Available: <http://www.openstack.org/foundation/companies/>.

[3] P. Mell και T. Grance, «<http://csrc.nist.gov>,» National Institute of Standards and Technology, [Ηλεκτρονικό].

Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.

[4] «<https://www.google.gr>,» [Ηλεκτρονικό].

Available: <https://www.google.gr>.

[5] «<https://www.dropbox.com>,» [Ηλεκτρονικό].

Available: <https://www.dropbox.com>.

[6] «<https://aws.amazon.com/ec2/>,» [Ηλεκτρονικό].

Available: <https://aws.amazon.com/ec2/>.

[7] «<http://www.cisco.com/c/en/us/solutions/data-center-virtualization/private-cloud/index.html#~Overview>,» [Ηλεκτρονικό].

Available: <http://www.cisco.com/c/en/us/solutions/data-center-virtualization/private-cloud/index.html#~Overview>.

[8] «<http://www.dell.com/learn/us/en/555/dell-cloud-computing?c=us&l=en&s=biz>,» [Ηλεκτρονικό].

Available: <http://www.dell.com/learn/us/en/555/dell-cloud->

computing?c=us&l=en&s=biz.

[9] «<http://www.ibm.com/cloud-computing/us/en/private-cloud.html>,» [Ηλεκτρονικό].

Available: <http://www.ibm.com/cloud-computing/us/en/private-cloud.html>.

[10] R. Hill, H. Laurie, L. Peter και M. Siavash, Guide to Cloud Computing Principles and Practice, 2013.

[11] «<http://www.ibm.com/cloud-computing/us/en/iaas.html>,» [Ηλεκτρονικό].

Available: <http://www.ibm.com/cloud-computing/us/en/iaas.html>.

[12] «<http://aws.amazon.com>,» [Ηλεκτρονικό].

Available: <http://aws.amazon.com>.

[13] «<http://www.bluelock.com>,» [Ηλεκτρονικό].

Available: <http://www.bluelock.com>.

[14] «<http://www.csc.com/cloud>,» [Ηλεκτρονικό].

Available: <http://www.csc.com/cloud>.

[15] «<http://www.rackspace.com>,» [Ηλεκτρονικό].

Available: <http://www.rackspace.com>.

[16] «<https://aws.amazon.com/elasticbeanstalk/>,» [Ηλεκτρονικό].

Available: <https://aws.amazon.com/elasticbeanstalk/>.

[17] «<http://azure.microsoft.com/en-us/>,» [Ηλεκτρονικό].

Available: <http://azure.microsoft.com/en-us/>.

[18] «<https://www.heroku.com>,» [Ηλεκτρονικό].

Available: <https://www.heroku.com>.

[19] «<http://www.salesforce.com/platform/overview/?d=70130000000lan8>,» [Ηλεκτρονικό].

Available: <http://www.salesforce.com/platform/overview/?d=70130000000lan8>.

[20] «<https://developers.google.com/appengine/?csw=1>,» [Ηλεκτρονικό].

Available: <https://developers.google.com/appengine/?csw=1>.

[21] «www.gmail.com,» [Ηλεκτρονικό].

Available: www.gmail.com.

[22] J. Atul, D. Johnson, M. Kiran, R. Murthy, C. Vivek και G. Yogesh, OpenStack Beginner's Guide v3.0, 2012.

[23] «<http://openstack.redhat.com>,» [Ηλεκτρονικό].

Available: <http://openstack.redhat.com/forum/discussion/577/howto-packstack-allinone-install-with-neutron-and-external-connectivity>.

Παραρτήματα

Παράρτημα Α

Πίνακας εικόνων

Εικόνα 1	Απλή Αρχιτεκτονική του OpenStack.....	78
Εικόνα 2	Αλληλεπίδραση μεταξύ του keystone και του horizon	78
Εικόνα 3	Υπηρεσία ταυτοποίησης Keystone	84

Εισαγωγή στο OpenStack και στα Components του

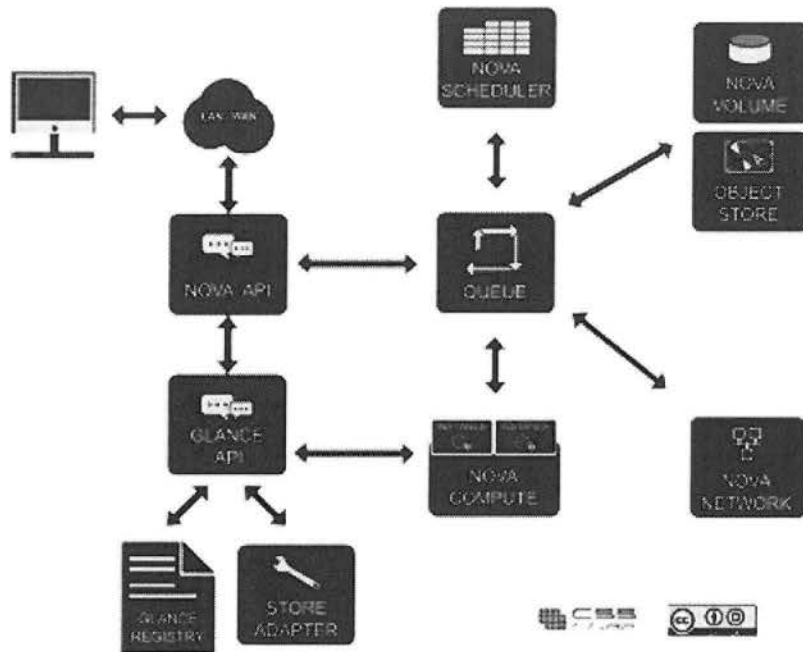
Όπως έχουμε αναφέρει το Openstack είναι ένα ανοιχτού λογισμικού έργο το οποίο μπορεί να χρησιμοποιηθεί από διάφορους παρόχους υπηρεσιών (επιχειρήσεις) και ιδιώτες. Δίνει την δυνατότητα να χρησιμοποιηθούν ως μια ενιαία υπηρεσία οι εικονικοί servers και ο αποθηκευτικός χώρος μαζί.

Πρόκειται για μια δημιουργία της Rackspace και της NASA. Η πρώτη βοήθησε στο κομμάτι του storage και η δεύτερη στο computing. Σήμερα το OpenStack θεωρείται η πιο διαδεδομένη υπηρεσία cloud με μεγάλες δυνατότητες επέκτασης(escalation).

Ας δούμε τα βασικότερα κομμάτια αρχιτεκτονικής του OPENSTACK.

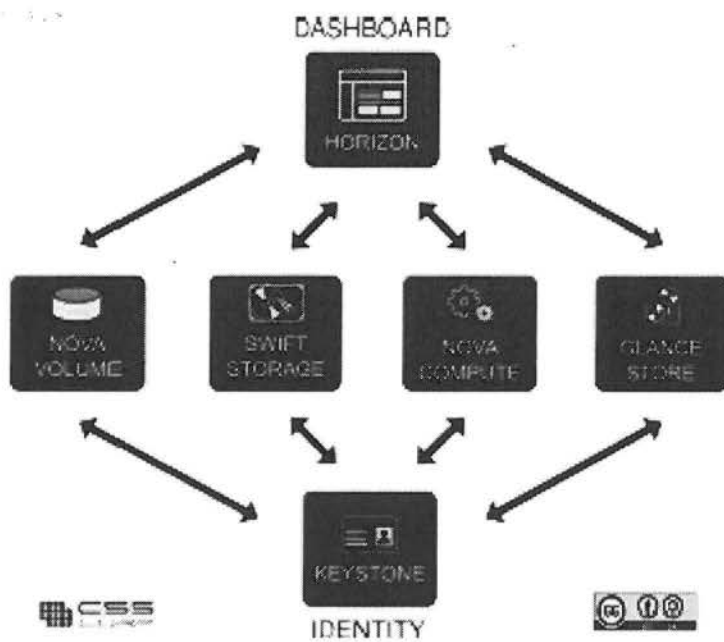
Οι κύριες “οικογένειες” υπηρεσιών (services) είναι οι εξής:

- Nova - Compute Service, το service που διαχειρίζεται τα εικονικά μηχανήματα
- Swift - Storage Service, το service που διαχειρίζεται το χώρο
- Glance - Imaging Service, το service που διαχειρίζεται τα images(iso κα)
- Keystone - Identity Service, το service που διαχειρίζεται την πιστοποίηση χρηστών
- Horizon - UI Service το service που διαχειρίζεται τη διεπαφή χρήστη



Εικόνα 1 Απλή Αρχιτεκτονική του OpenStack

Η παρακάτω δείχνει τη αλληλεπίδραση μεταξύ του keystone και του horizon



Εικόνα 2 Αλληλεπίδραση μεταξύ του keystone και του horizon

ΥΠΟΔΟΜΗ NOVA

Η nova είναι ο ελεγκτής θα μπορούσαμε να πούμε για την υπολογιστική σύννεφου(cloud). Όλες οι δραστηριότητες που αφορούν τον κύκλο ζωής των στιγμιότυπων(εικονικές μηχανές) χειρίζονται από τη nova. Είναι κατά κάποιο τρόπο η πλατφόρμα που διαχειρίζεται τους πόρους του computing(networking authorization).Βέβαια δεν μπορεί να κατασκευάσει EM από μόνη της αλλά μέσω libvirt API επικοινωνεί με hypervisors. Η nova είναι συμβατή με το EC2 API της Amazon Web Services.

Λειτουργίες και Χαρακτηριστικά:

- Διαχείριση κύκλου ζωής στιγμιότυπων
- Διαχείριση υπολογιστικών πόρων
- Δικτύωση και Πιστοποίηση
- API βασισμένο σε REST
- Ασύγχρονη επικοινωνία
- Hypervisor : Υποστηρίζει Xen, XenServer/XCP, KVM, UML, VMware vSphere και Hyper-V

Στοιχεία του OpenStack Compute

- 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)

Το API Server παρέχει μία διεπαφή για την επικοινωνία του “έξω κόσμου” με την cloud υποδομή. Ο API server είναι το μόνο στοιχείο το οποίο χρησιμοποιείται για τη διαχείριση της υποδομής.

Αυτό επιτυγχάνεται μέσω κλήσεων των web services χρησιμοποιώντας EC2 API.

Η επικοινωνία μεταξύ των components επιτυγχάνεται με τον server Message Queue. Ως εναλλακτική στο EC2 API, το OpenStack παρέχει επίσης ένα δικό της API ονομαζόμενο "OpenStack API".

Message Queue (Rabbit MQ Server)

Το OpenStack επικοινωνεί μεταξύ των στοιχείων χρησιμοποιώντας το message queue μέσω AMQP(Advanced Message Queue Protocol). Η Nova χρησιμοποιεί ασύγχρονες κλήσεις για αίτηση απόκριση, με μία επανάκληση οποία ξεκινάει όταν η απόκριση λαμβάνεται. Με αυτόν τον τρόπο δεν έχουμε μεγάλη καθυστέρηση όταν απαιτείται να εξυπηρετηθούν πολλές API κλήσεις ταυτόχρονα.

Compute Worker (nova-compute)

Οι Compute workers παίρνουν τα αιτήματα από το message queue και διεκπεραιώνουν τις λειτουργίες . Σε μια μεγάλη εγκατάσταση υπάρχουν πολλοί τέτοιοι. Ένα instance αναπτύσσεται σε οποιοδήποτε από τα διαθέσιμα compute worker με βάση το scheduling algorithm που χρησιμοποιείται.

Network Controller (nova-network)

Το Network Controller σχετίζεται με τη ρύθμιση του δικτύου των vm από τον host . Δεσμεύει IP διευθύνσεις, ρυθμίζοντας VLANs για projects, υλοποιώντας security groups και ρυθμίζοντας δίκτυα για τα compute nodes.

Volume Workers (nova-volume)

Volume workers χρησιμοποιείται για τη διαχείριση των LVM-based instance volumes. Τελούν λειτουργίες όπως η δημιουργία η διαγραφή η προσάρτησή του σε ένα instance, και η αποπροσάρτησή του. Τα Volumes παρέχουν ένα τρόπο για μόνιμο storage για τα instances, αφού το root partition δεν είναι τέτοιο μιας και διαγράφοντας ένα instance διαγράφεται και αυτό.

Μπορεί να χρησιμοποιηθεί για database servers κα.

Scheduler (nova-scheduler)

Ο προγραμματιστής χαρτογραφεί τις κλήσεις nova-API για τα κατάλληλα εξαρτήματα OpenStack. Τρέχει σαν δαίμονας που ονομάζεται nova-schedule και

επιλέγει ένα compute server από ένα σύνολο διαθέσιμων πόρων, ανάλογα με τον αλγόριθμο προγραμματισμού που τρέχει. Ένας προγραμματιστής-scheduler μπορεί να

βασίζει τις αποφάσεις του σε διάφορους παράγοντες, όπως το φορτίο, τη μνήμη, τη φυσική απόσταση της ζώνης διαθεσιμότητας, την αρχιτεκτονική CPU, κλπ.

Αυτή τη στιγμή το nova-scheduler υλοποιεί μερικούς βασικούς αλγορίθμους προγραμματισμού:

- chance: με αυτή τη μέθοδο ο compute host επιλέγεται τυχαία μέσω των διαθέσιμων ζωνών.
- availability zone: Ίδιο με το προηγούμενο , αλλά μέσω καθορισμένης ζώνης.
- simple: Με αυτή τη μέθοδο το node με το ελάχιστο φορτίο φτιάχνει το instance. Οι πληροφορίες λαμβάνονται απο load balancer.

OpenStack Imaging Service (Glance)

OpenStack Imaging Service είναι ένα service το οποίο είναι υπεύθυνο για τα virtual machine images. Μπορεί να ρυθμιστεί ώστε να χρησιμοποιεί τα ακόλουθα storage backends:

- Local filesystem (default)
- OpenStack Object Store για αποθήκευση των images
- S3 storage απευθείας
- S3 storage με Object Store ως ενδιάμεσο για πρόσβαση στον S3.
- HTTP (read-only)

Χαρακτηριστικά και Λειτουργίες

- Παρέχει υπηρεσίες για τα images

Components του Glance

- Glance-control
- Glance-registry

Υποδομή αποθήκευσης OpenStack (Swift)

Το Swift παρέχει ένα κατανεμημένο , τελικά συνεπές εικονικό αντικείμενο αποθήκευσης για το OpenStack. Είναι αντίστοιχο με το Amazon Web Services -

Simple Storage Service (S3). Το Swift μπορεί να αποθηκεύσει δισεκατομμύρια αντικειμένων κατανεμημένα μέσα στα nodes. Το Swift έχει φτιαχτεί έχοντας ληφθεί υπόψη η διαχείριση πλεονασμού πόρων (redundancy) και failover και μπορεί να κάνει αρχειοθέτηση και media streaming . Είναι υπερβολικά επεκτάσιμο και σε χώρο (petabytes) και σε αριθμό αντικειμένων που αποτελούν το storage.

Λειτουργίες και χαρακτηριστικά γνωρίσματα

- Αποθήκευση μεγάλου αριθμού αντικειμένων
- Αποθήκευση του μεγάλου μεγέθους αντικειμένων
- Data Redundancy
- Δυνατότητες Αρχειακές - Εργασία με μεγάλα σύνολα δεδομένων
- Δοχείο δεδομένων(data container) για εικονικές μηχανές και cloud εφαρμογές
- Δυνατότητες ροής πολυμέσων
- Ασφαλής αποθήκευση αντικειμένων
- Δημιουργία αντιγράφων ασφαλείας και αρχειοθέτηση
- Υπερβολική επεκτασιμότητα

Συνιστώσες του Swift

- Swift Λογαριασμός
- Swift Container
- Swift Object
- Swift Proxy
- Το RING

Swift Proxy Server

Οι καταναλωτές-consumers αλληλοεπιδρούν με το στήσιμο του Swift μέσω του διακομιστή μεσολάβησης χρησιμοποιώντας το Swift API. Ο διακομιστής μεσολάβησης λειτουργεί ως φύλακας και παραλαμβάνει τις αιτήσεις από τον έξω κόσμο . Αυτομάτως αναζητά τη θέση των κατάλληλων φορέων και των διαδρομών των αιτήσεων τους .

Ο διακομιστής μεσολάβησης χειρίζεται επίσης τις αποτυχίες των οντοτήτων με αναδρομολόγηση των αιτήσεων failover φορείς (φορείς handoff)

Swift Object Server

Ο διακομιστής Object είναι σαν άμορφη μάζα . Ευθύνη έχει να χειριστεί την αποθήκευση, ανάκτηση και τη διαγραφή των αντικειμένων που αποθηκεύονται στο τοπικό μέσο αποθήκευσης . Τα αντικείμενα είναι συνήθως δυαδικά αρχεία που είναι αποθηκευμένα στο σύστημα αρχείων με μεταδεδομένα που περιέχονται ως “extended file attributes (xattr)”.

Σημείωση : το xattr υποστηρίζεται σε αρκετά συστήματα αρχείων , όπως ext3 , ext4 , XFS , Btrfs , JFS και ReiserFS στο Linux . Αλλά είναι γνωστό ότι δουλεύει πιο σωστά καλύτερα κάτω από XFS , JFS , ReiserFS , reiser4 και ZFS . Το XFS θεωρείται ότι είναι η καλύτερη επιλογή.

Swift Container Server

Ο διακομιστής Container απαριθμεί τα αντικείμενα σε ένα δοχείο. Οι πίνακες αποθηκεύονται ως αρχεία SQLite . Ο διακομιστής Container παρακολουθεί επίσης τις

στατιστικές, όπως τον αριθμό των αντικειμένων που περιέχονται και το μέγεθος αποθήκευσης που καταλαμβάνεται από ένα Container.

Swift Account Server

Ο διακομιστής του λογαριασμού απαριθμεί τα εμπορευματοκιβώτια με τον ίδιο τρόπο ένας διακομιστής δοχείο απαριθμεί τα αντικείμενα .

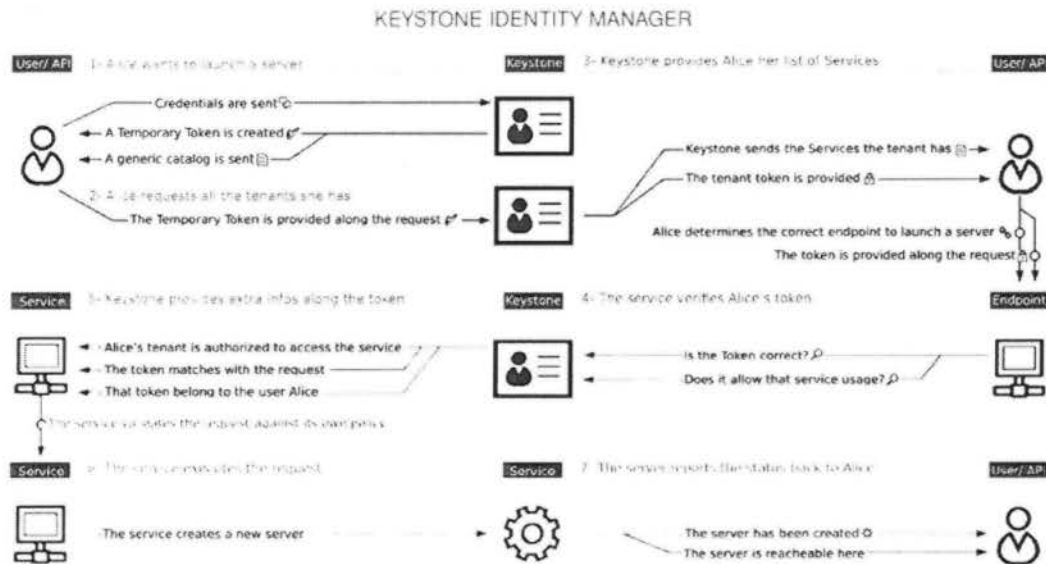
To Ring

Το Ring περιέχει πληροφορίες σχετικά με τη φυσική θέση των αντικειμένων που αποθηκεύονται μέσα στο Swift. Είναι μια εικονική αναπαράσταση της χαρτογράφησης των ονομάτων των φορέων για την πραγματική φυσική τους θέση. Είναι ανάλογο με μια υπηρεσία ευρετηρίου που χρησιμοποιούν διάφορες μεθόδους για να αναζήτηση και να εντοπίσετε την πραγματική φυσική θέση των οντοτήτων μέσα στη συστάδα. Οντότητες, όπως λογαριασμοί, εμπορευματοκιβώτια, τα αντικείμενα έχουν τα δικά ξεχωριστά δαχτυλίδια.

OpenStack Υπηρεσία Identity (Keystone)

Το Keystone παρέχει υπηρεσίες ταυτοποίησης και πρόσβασης για όλα τα στοιχεία στην οικογένεια OpenStack. Υλοποιεί το δικό REST-based API . Παρέχει επαλήθευση ταυτότητας και αδειοδότηση για όλα τα συστατικά του OpenStack, συμπεριλαμβανομένων των Glance,

Swift, Nova. Ο έλεγχος ταυτότητας επαληθεύει ότι η αίτηση προέρχεται στην πραγματικότητα από εκεί που λέει. Εξουσιοδότηση είναι η επαλήθευση αν ο πιστοποιημένος χρήστης έχει πρόσβαση στις υπηρεσίες που αυτός / αυτή αιτούνται.



Εικόνα 3 Υπηρεσία ταυτοποίησης Keystone

Το Keystone παρέχει δύο τρόπους ελέγχου ταυτότητας. Ένα είναι το όνομα χρήστη / κωδικό πρόσβασης που βασίζεται και το άλλο είναι token based. Εκτός αυτού το keystone παρέχει τις ακόλουθες υπηρεσίες:

- Token Υπηρεσία (που μεταφέρει πληροφορίες εξουσιοδότησης για έναν πιστοποιημένο χρήστη)
- Υπηρεσία καταλόγου (το οποίο περιέχει μια λίστα με τις διαθέσιμες υπηρεσίες στη διάθεση των χρηστών)
- Υπηρεσία Πολιτικής (Δίνει τη δυνατότητα στο keystone να διαχειρίζεται την πρόσβαση σε συγκεκριμένες υπηρεσίες από συγκεκριμένους χρήστες ή ομάδες).

Εξαρτήματα της Υπηρεσίας ταυτότητας

- Endpoints -

Κάθε υπηρεσία OpenStack (Nova, Swift, ματιά) τρέχει σε μια ειδική θύρα και σε ένα ειδικό URL (host), τα οποία ονομάζουμε EndPoints.

- Regions

- Μια περιοχή region καθορίζει μια ειδική φυσική τοποθεσία μέσα σε ένα κέντρο δεδομένων. Σε μία τυπική εγκατάσταση σύννεφο, οι περισσότερες αν όχι όλες οι υπηρεσίες κατανέμονται σε κέντρα δεδομένων / servers που ονομάζονται επίσης regions.

- User – Ένας χρήστης πιστοποιημένος από το keystone.

- Services – Κάθε συνιστώσα-component στην οποία συνδέεται ή διαχειρίζεται από το keystone μπορεί να ειπωθεί service. Για παράδειγμα λέμε Glance ένα keystone service.

- Role - Προκειμένου να διατηρήσουν τους περιορισμούς ως προς το τι ένας συγκεκριμένος χρήστης μπορεί να κάνει μέσα σε υποδομές cloud, είναι σημαντικό να έχουμε ένα σχετιζόμενο με αυτό ρόλο.

- Tenant - A tenant είναι ένα project με το οποίο σχετίζονται όλα τα service endpoint και ένας ρόλος που σχετίζεται με χρήστη ο οποίος είναι μέλος στο συγκεκριμένο tenant.

Διαχείριση του Openstack μέσω Web-Interface (Horizon)

Η web διεπαφή Horizon είναι ένας πιο εύκολος και γραφικός τρόπος για να διαχειριστείς τις OpenStack υπηρεσίες. Ο χρήστης μπορεί μέσα από αυτήν να διαχειριστεί τα instances, τα images, να δημιουργήσει κλειδιά , να προσαρτήσει volumes στα instances, να διαμορφώσει τα Swift containers κτλ. Μέσα από το γραφικό περιβάλλον στο web γίνονται τα πάντα. Επίσης επιτρέπει την πρόσβαση των χρηστών σε κονσόλα και μπορεί να συνδεθεί μέσω ενσωματωμένου VNC.

Συνολικά, το Horizon διαθέτει τα ακόλουθα:

Διαχείριση

- στα Instances – Δημιουργία , καταστροφή , δυνατότητα να ελεγχθούν τα logs και σύνδεση μέσω VNC, προσάρτηση volumes, κτλ.

- πρόσβασης και ασφάλειας - Δημιουργία security groups, διαχείριση κλειδιών, ανάθεση floating IPs, κτλ.

- Flavor - Διαχείριση των δυνατών προδιαγεγραμμένων μεθόδων για την κατασκευή ενός instance , κάτι σαν εικονικά hardware προτύπων.

- Image - Επεξεργασία ή διαγραφή τους.

- Παρακολούθηση των υπηρεσιών.

- Παρακολούθηση χρηστών και επίπεδο χρησιμοποίησης των projects.
- Χρήστη – Δημιουργία Διαγραφή κτλ.
- Των Volume - Δημιουργία τους και κατασκευή στιγμιότυπων τους για μελλοντική χρήση.
- Του Object Store - Δημιουργία ,διαγραφή των container και των objects.
- Κατέβασμα μεταβλητών συστήματος για ένα έργο.