



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

Τμήμα Ψηφιακών Συστημάτων

Ψηφιακές Επικοινωνίες και Δίκτυα

Διπλωματική εργασία

Πειραματική μελέτη και αξιολόγηση της χρήσης του
Openstack ως εργαλείου για την διαχείριση υποδομών
υπολογιστικού νέφους

Όνομα / επώνυμο: Κωνσταντίνος - Λυκούργος Μαρίνης

Αριθμός μητρώου: ΜΕ 14038

Επιβλέπων καθηγητής: Άγγελος Ρούσκας

Πειραιάς, 2017

Πίνακας περιεχομένων

Περίληψη.....	5
1. Cloud computing - Εισαγωγή	5
1.1 Cloud Computing ή Cloud - Προέλευση του όρου	5
1.2 Κυριότερα μοντέλα Cloud Computing	6
1.3 Βασικά χαρακτηριστικά - Πλεονεκτήματα	8
1.4 Ακεραιότητα Δεδομένων – Ασφάλεια	9
1.5 Εμπιστευτικότητα των δεδομένων.....	11
1.5.1 Κίνδυνοι στο σύννεφο	12
1.5.2 Πλεονεκτήματα Ασφαλείας στο Cloud computing	13
1.5.3 Μέτρα προστασίας.....	14
1.6 Cloud computing, Private, Public ή και τα δύο;	14
1.6.1 Private Cloud, πλεονεκτήματα και μειονεκτήματα.....	15
1.6.2 Public Cloud, πλεονεκτήματα και μειονεκτήματα	15
1.7 Hybrid cloud	15
1.8 Οι μεγαλύτεροι πάροχοι Cloud computing.....	16
▪ Amazon Web Services	16
▪ Microsoft	16
▪ IBM	16
▪ Google	17
▪ Salesforce	17
2. Hypervisors.....	17
2.1 Τύποι Hypervisor	18
3. OpenStack.....	19
3.1 Εισαγωγή στο OpenStack	19
3.1.1 OpenStack Foundation	19
3.2 Αρχιτεκτονική	20
3.2.1 Controller.....	20
3.2.2 Compute	21
3.2.3 Block Storage	21
3.2.4 Object Storage.....	21
3.3 Τα κύρια μέρη του OpenStack	21
▪ Nova.....	22
▪ Swift.....	22
▪ Cinder	22
▪ Neutron	22
▪ Horizon	22
▪ Keystone.....	22
▪ Glance.....	22
▪ Ceilometer	23
▪ Heat	23
3.4 Πώς χρησιμοποιείται το OpenStack σε Cloud περιβάλλον;.....	23
4. Εγκατάσταση και διαμόρφωση OpenStack.....	24
4.1 Διαμόρφωση name resolution	25

4.2 Networking	28
4.2.1 Επαλήθευσης σύνδεσης των κόμβων	31
4.3 Network Time Protocol (NTP).....	34
4.3.1 Επαλήθευση λειτουργίας του NTP.....	36
4.4 OpenStack packages.....	37
4.4.1 Ενεργοποίηση του OpenStack repository	38
4.4.2 Ολοκλήρωση εγκατάστασης	38
4.5 Database.....	38
4.5.1 Εγκατάσταση πακέτων στον Controller node	39
4.5.2 Ολοκλήρωση εγκατάστασης	39
4.6 Διακομιστής μηνυμάτων - Messaging server.....	40
4.6.1 Εγκατάσταση και διαμόρφωση	41
4.7 Memcached.....	42
4.7.1 Εγκατάσταση και διαμόρφωση	42
4.7.2 Ολοκλήρωση εγκατάστασης	42
4.8 Υπηρεσία ταυτότητας - Identity service.....	43
4.8.1 Έννοιες OpenStack Identity Service.....	43
4.8.2 Η υπηρεσία "Identity service" περιέχει τα παρακάτω στοιχεία	44
4.8.3 Εγκατάσταση και διαμόρφωση	45
4.8.4 Ολοκλήρωση εγκατάστασης	48
4.8.5 Επαλήθευση λειτουργίας.....	50
4.9 Image service.....	53
4.9.1 Εγκατάσταση και διαμόρφωση.....	53
4.9.2 Επαλήθευση λειτουργίας.....	61
4.10 Compute service.....	62
4.10.1 Το OpenStack Compute αποτελείται από τα ακόλουθα.....	63
4.10.2 Εγκατάσταση και διαμόρφωση – Controller node	63
4.10.3 Εγκατάσταση και διαμόρφωση – Compute node	71
4.10.4 Ολοκλήρωση εγκατάστασης	75
4.10.5 Επαλήθευση λειτουργίας – Controller node.....	75
4.11 Networking service.....	76
4.11.1 Εγκατάσταση και διαμόρφωση Controller node	76
4.11.2 Configure networking options.....	78
4.11.3 Ολοκλήρωση εγκατάστασης	89
4.11.4 Εγκατάσταση και διαμόρφωση – Compute node	90
4.11.6 Ολοκλήρωση εγκατάστασης	94
4.12 Dashboard	95
4.12.1 Εγκατάσταση και διαμόρφωση.....	95
4.12.2 Ολοκλήρωση εγκατάστασης	98
4.13 Block Storage service.....	100
4.13.1 Εγκατάσταση και διαμόρφωση.....	101
4.13.2 Εγκατάσταση και διαμόρφωση Controller node	104
4.13.3 Ολοκλήρωση εγκατάστασης	106
4.13.4 Εγκατάσταση και διαμόρφωση Block Storage node.....	107

4.13.5 Ολοκλήρωση εγκατάστασης	109
4.13.6 Επαλήθευση λειτουργίας	109
4.14 Launch an instance	110
4.14.2 Launch the instance on the provider network	113
4.14.3 Status of instance	114
4.14 Object Storage service.....	116
4.14.1 Εγκατάσταση και διαμόρφωση - Controller node	117
4.14.2 Εγκατάσταση και διαμόρφωση components.....	119
4.14.3 Εγκατάσταση και διαμόρφωση Storage node -ObjectStorage1, ObjectStorage2-....	121
4.14.4 Εγκατάσταση και διαμόρφωση components.....	124
4.14.5 Δημιουργία και διανομή initial rings.....	128
4.14.6 Επαλήθευση λειτουργίας.....	132
4.15 Orchestration service	134
4.15.1 Εγκατάσταση και διαμόρφωση.....	134
4.15.2 Δημιουργία Orchestration service API endpoints.....	136
4.15.3 Εγκατάσταση και διαμόρφωση components.....	139
4.15.4 Ολοκλήρωση εγκατάστασης	141
4.15.5 Επαλήθευση λειτουργίας.....	142
4.15.6 Δημιουργία stack.....	142
Συμπέρασμα.....	144
Βιβλιογραφία - Αναφορές.....	147

Πίνακας εικόνων

FIGURE 1: CLOUD COMPUTING	6
FIGURE 2: TYPES OF CLOUD COMPUTING	6
FIGURE 3: INFRASTRUCTURE AS A SERVICE – IAAS	7
FIGURE 4: PLATFORMS AS A SERVICE - PAAS	7
FIGURE 5: SOFTWARE AS A SERVICE - SAAS	7
FIGURE 6: ORGANIZATION OF DATA SECURITY AND PRIVACY IN CLOUD COMPUTING.....	10
FIGURE 7: CLOUD COMPUTING SECURITY.....	13
FIGURE 8: TYPE 1 HYPERVISOR.....	18
FIGURE 9: TYPE 2 HYPERVISOR.....	18
FIGURE 10 : OPENSTACK CLOUD SOFTWARE	19
FIGURE 11 : OPENSTACK NEWTON HARDWARE REQUIREMENTS	20
FIGURE 12: OPENSTACK DASHBOARD	98
FIGURE 13: OPENSTACK DASHBOARD	99
FIGURE 14: OPENSTACK DASHBOARD	99
FIGURE 15: OPENSTACK DASHBOARD	99
FIGURE 16: INSTANCES.....	114
FIGURE 17: NETWORK TOPOLOGY & GRAPH	114
FIGURE 18: TEST-VM1	115
FIGURE 19: TEST-VM2	116
FIGURE 20: CONTAINER1.....	133
FIGURE 21: CONTAINER DEMO	133
FIGURE 22: STACK.....	143
FIGURE 23: STACK TOPOLOGY.....	143
FIGURE 24: SECURITY GROUP RULES.....	146

Περίληψη

Σκοπός της παρούσας εργασίας είναι η δημιουργία μιας εφαρμογής Cloud computing μέσω μιας open-source πλατφόρμας OpenStack που αναπτύσσεται κυρίως ως Infrastructure-as-a-service (IaaS) και η εγκατάσταση υπηρεσιών σε αυτήν.

Το OpenStack κατασκευάζεται από μια ευημερούσα κοινότητα προγραμματιστών, σε συνεργασία με τους χρήστες και έχει την δυνατότητα να ελέγχει μεγάλες ομάδες υπολογιστικών πόρων, αποθήκευσης και δικτύου σε ένα Data Center το οποίο διαχειρίζεται μέσω ενός πίνακα ελέγχου (dashboard) ή μέσω του API OpenStack. Το OpenStack λειτουργεί με δημοφιλείς τεχνολογίες ανοιχτού κώδικα, καθιστώντας το ιδανικό για ετερογενή υποδομή. Πολλές επιχειρήσεις παγκοσμίως βασίζονται στο OpenStack για να λειτουργούν καθημερινά μειώνοντας το κόστος και βοηθώντας τους να κινηθούν γρηγορότερα.

Επιπλέον θα αναλυθεί, ο όρος Cloud computing, η ασφάλεια του, οι διάφοροι τύποι Cloud και οι υπηρεσίες που μπορούν να προσφερθούν στον τελικό χρήστη.

Τέλος θα γίνει αναφορά στις δυσκολίες και στα σφάλματα που προέκυψαν κατά την διάρκεια υλοποίησης της εφαρμογής σε περιβάλλον Ubuntu Server και οι τρόποι αντιμετώπισης τους.

1. Cloud computing - Εισαγωγή

"Cloud Computing" ή "Cloud " ονομάζεται ένας Internet-based computing τύπος που παρέχει κοινή επεξεργασία πόρων αν και όταν ζητηθεί (on-demand) σε όλες τις συσκευές που είναι συνδεδεμένες στο Internet όπως δίκτυα υπολογιστών, Servers, Storage, εφαρμογές και υπηρεσίες, τα οποία μπορούν να τροφοδοτηθούν εύκολα και γρήγορα με ελάχιστη ικανότητα διαχείρισης. Το "Cloud Computing" και οι λύσεις αποθήκευσης παρέχονται σε χρήστες και επιχειρήσεις με διάφορες δυνατότητες για αποθήκευση και επεξεργασία των δεδομένων τους είτε από τους ίδιους, είτε από τρίτους συνεργάτες που βρίσκονται είτε στην ίδια πόλη, είτε στην άλλη μεριά του πλανήτη. Το "Cloud Computing" βασίζεται στην σωστή κατανομή των πόρων όπως συμβαίνει και σε ένα δίκτυο ηλεκτρικής ενέργειας. (Wikipedia, n.d.)

1.1 Cloud Computing ή Cloud - Προέλευση του όρου

Η προέλευση του Cloud Computing δεν είναι ξεκάθαρη. Η λέξη «Cloud » χρησιμοποιείται για να περιγράψει μεγάλη συσσώρευση αντικειμένων που εμφανίζονται οπτικά από μια απόσταση ως σύννεφο (Cloud) και δεν επιθεωρούνται περαιτέρω σε ένα δεδομένο πλαίσιο. Μια άλλη εξήγηση είναι ότι τα παλαιά προγράμματα που έφεραν σχηματικές αναπαραστάσεις δικτύων περιέβαλε τις εικόνες για τους κεντρικούς υπολογιστές με έναν κύκλο, και μια συστάδα των κεντρικών υπολογιστών σε ένα διάγραμμα δικτύων είχε διάφορους επικαλύπτοντας κύκλους, που έμοιαζαν με σύννεφο.[16] στην αναλογία στην ανωτέρω χρήση, η λέξη "Cloud " χρησιμοποιήθηκε ως μεταφορά για το Διαδίκτυο και μια τυποποιημένη νεφελοειδής μορφή χρησιμοποιήθηκε για να δείξει ένα δίκτυο στις σχηματικές αναπαραστάσεις τηλεφωνίας. Αργότερα χρησιμοποιήθηκε για να απεικονίσει το Διαδίκτυο στα διαγράμματα δικτύων υπολογιστών. (Wikipedia, n.d.)

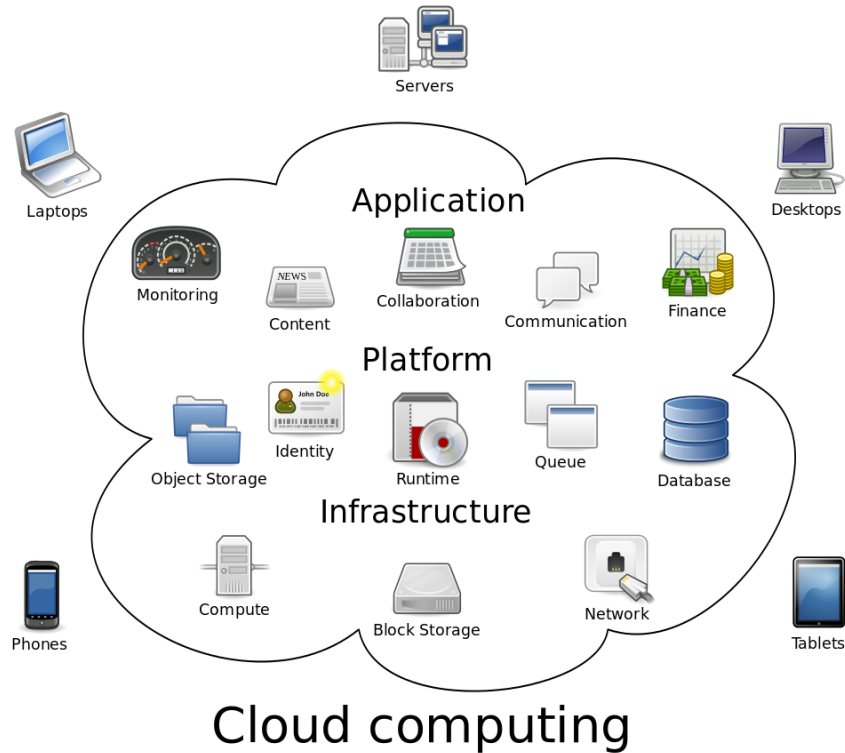


Figure 1: Cloud computing

1.2 Κυριότερα μοντέλα Cloud Computing

Το Cloud computing έχει τρία κύρια είδη που αναφέρονται συνήθως ως Infrastructure as a Service (IaaS), Platform as a Service (PaaS), και Software as a Service (SaaS).



Figure 2: Types of Cloud Computing

➤ Infrastructure as a Service (IaaS)

Η Υποδομή ως υπηρεσία - Infrastructure as a Service, IaaS, περιέχει τα βασικά δομικά στοιχεία για το Cloud και τυπικά παρέχει πρόσβαση σε δυνατότητες δικτύωσης, υπολογιστές (virtual or on dedicated hardware), καθώς και χώρο αποθήκευσης δεδομένων. Η Υποδομή ως υπηρεσία - Infrastructure as a Service, IaaS παρέχει το υψηλότερο επίπεδο ευελιξίας και ελέγχου της διαχείρισης των πόρων που είναι παρόμοια με την υπάρχουσα υποδομή των πόρων που πολλά τμήματα IT και προγραμματιστές είναι εξοικειωμένοι σήμερα. (Amazon IaaS, PaaS, SaaS, n.d.)

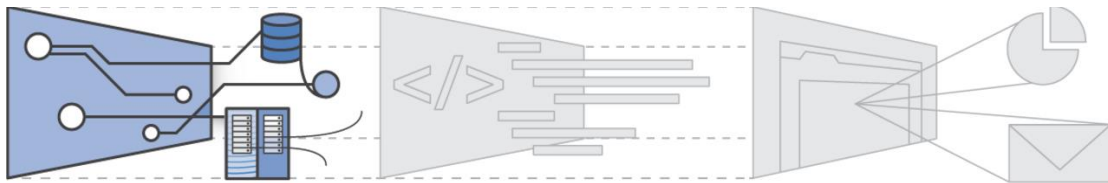


Figure 3: Infrastructure as a Service – IaaS

➤ Platform as a Service (PaaS)

Οι Πλατφόρμες ως υπηρεσία - Platforms as a service, PaaS καταργούν την ανάγκη των επιχειρήσεων να διαχειριστούν την υποκείμενη υποδομή (συνήθως hardware και λειτουργικά συστήματα) και επιτρέπει να επικεντρωθούν στην ανάπτυξη και τη διαχείριση των εφαρμογών τους. Αυτό βοηθά ώστε να είναι πιο αποτελεσματική και να μην χρειάζεται να ανησυχούν για την διακύμανση των πόρων, την ικανότητα σχεδιασμού, τη συντήρηση λογισμικού, την επιδιόρθωση, ή οποιοδήποτε άλλο πρόβλημα που μπορεί να έχει επίπτωση στη λειτουργία της εφαρμογής τους. (Amazon IaaS, PaaS, SaaS, n.d.)

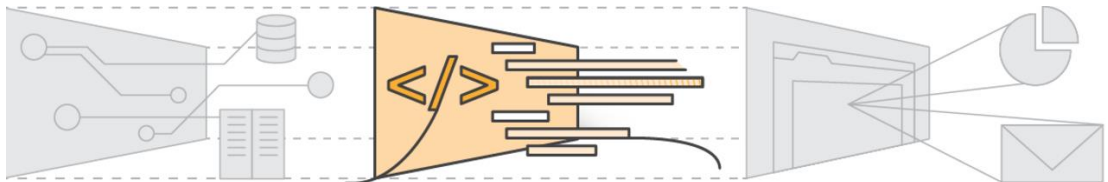


Figure 4: Platforms as a service - PaaS

➤ Software as a Service (SaaS)

Το Λογισμικό ως Υπηρεσία - Software as a Service, SaaS παρέχει ένα ολοκληρωμένο προϊόν που διαχειρίζεται από τον πάροχο αυτής της υπηρεσίας. Στις περισσότερες περιπτώσεις, οι άνθρωποι ως Software as a Service αναφέρονται στις end-user εφαρμογές. Με το SaaS δεν χρειάζεται να σκεφτεί ο τελικός χρήστης για το πώς η υπηρεσία διατηρείται ή πώς γίνεται η διαχείριση της σχετικής υποδομής! το μόνο που χρειάζεται να σκεφτεί είναι για το πώς θα χρησιμοποιήσει την συγκεκριμένη υπηρεσία. Ένα κοινό παράδειγμα μιας εφαρμογής SaaS είναι το ηλεκτρονικό ταχυδρομείο όπου στέλνουμε και λαμβάνουμε e-mail, χωρίς να χρειάζεται να διαχειριστούμε οτιδήποτε εκτός απ' τον λογαριασμό μας. (Amazon IaaS, PaaS, SaaS, n.d.)

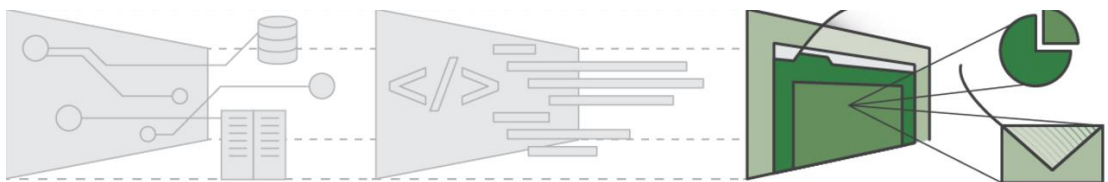


Figure 5: Software as a Service - SaaS

1.3 Βασικά χαρακτηριστικά - Πλεονεκτήματα

- **Flexibility – Ευελιξία**

Οι Cloud -based υπηρεσίες είναι ιδανικές για τις επιχειρήσεις με κυμαινόμενες ανάγκες εύρους ζώνης. Αν αυξηθούν οι ανάγκες είναι εύκολο να αναβαθμιστεί η ικανότητα του "Cloud", αντλώντας από απομακρυσμένους διακομιστές τους πόρους. Παρομοίως, αν μειωθεί η ζήτηση μπορούν και πάλι να μειωθούν. Αυτό το επίπεδο ευελιξίας μπορεί να δώσει στις επιχειρήσεις που χρησιμοποιούν το Cloud computing ένα πραγματικό πλεονέκτημα έναντι των ανταγωνιστών.

- **Disaster recovery - Ανάκτηση δεδομένων**

Οι επιχειρήσεις όλων των μεγεθών θα πρέπει να επενδύουν σε ανάκτηση καταστροφής δεδομένων, αλλά και για τις μικρότερες επιχειρήσεις που δεν έχουν την απαιτούμενη εμπειρία και το οικονομικό υπόβαθρο, αυτό είναι συχνά πιο ιδανικό από την πραγματικότητα. Το Cloud βοηθά σήμερα περισσότερες επιχειρήσεις να στραφούν προς αυτή την τάση. Σύμφωνα με την Ομάδα του Αμπερντίν, οι μικρές επιχειρήσεις έχουν διπλάσιες πιθανότητες από τις μεγαλύτερες εταιρείες να έχουν εφαρμόσει Cloud - based backup και μεθόδους ανάκτησης που εξοικονομούν χρόνο και χρήμα.

- **Αυτόματες ενημερώσεις λογισμικού**

Σημαντικό χαρακτηριστικό του Cloud computing είναι ότι οι servers δεν είναι στον χώρο των επιχειρήσεων και κατά συνέπεια οι προμηθευτές αναλαμβάνουν την συντήρηση τους αντί αυτών. Οι ενημερώσεις λογισμικού - συμπεριλαμβανομένων των ενημερώσεων ασφαλείας - γίνονται από τους παρόχους και δεν χρειάζεται να ανησυχεί κανένας χρήστης, αφήνοντας παράλληλα ελεύθερο χρόνο να επικεντρωθεί στα πράγματα που έχουν σημασία, όπως και την ανάπτυξη της επιχείρησής.

- **Κόστος**

Το Cloud computing μειώνει το υψηλό κόστος του υλικού. Με την μέθοδο "pay as you go" οι χρήστες ή επιχειρήσεις μπορούν να έχουν το μοντέλο που θέλουν βάση της συνδρομής τους. Επίσης αν στο παραπάνω προσθέσουμε και την ευκολία εγκατάστασης και διαχείρισης το κάνει πολύ φιλικό προς αυτούς.

- **Αύξηση συνεργασίας**

Όταν οι χρήστες μπορούν να έχουν πρόσβαση, να επεξεργαστούν και να μοιραστούν έγγραφα οποιαδήποτε στιγμή, από οπουδήποτε, είναι σε θέση να κάνουν περισσότερα πράγματα μαζί και με καλύτερα αποτελέσματα. Οι Cloud-based εφαρμογές βοηθάνε να κάνουν ενημερώσεις σε πραγματικό χρόνο και παρέχουν πλήρη προβολή των δεδομένων - αποτελεσμάτων τους.

- **Εργασία από οπουδήποτε**

Με το Cloud computing, χρειάζεται να υπάρχει μόνο μια σύνδεση στο Internet για να δουλέψει ο χρήστης, προσθέτοντας και κάποιες υπηρεσίες Cloud που προσφέρονται σε όλες τις φορητές συσκευές - smartphones, δεν υπάρχει κανένας περιορισμός.

Αυτό έχει ως αποτέλεσμα οι επιχειρήσεις να μπορούν να προσφέρουν πιο ευέλικτους τρόπους εργασίας για τους εργαζομένους, ώστε να μπορεί υπάρξει ισορροπία εργασίας-προσωπικής ζωής με την παραγωγικότητα που απαιτούν.

- **Document control - Έλεγχος εγγράφων**

Οι περισσότεροι χρήστες και συνεργάτες ανταλλάζουν έγγραφα και η ανάγκη για τον έλεγχο τους αυξάνεται. Πριν από το Cloud , οι εργαζόμενοι έπρεπε να ανταλλάξουν τα αρχεία τους αρκετές φορές ως συνημμένα ηλεκτρονικού ταχυδρομείου για να επεξεργαστούν από ένα χρήστη κάθε φορά με αποτέλεσμα την λάθος μορφοποίηση των αρχείων. Με αυτό τον τρόπο επιτυγχάνεται μεγαλύτερη διαφάνεια σε όλα τα αρχεία που είναι αποθηκευμένα σε κεντρικό επίπεδο και ο καθένας μπορεί να τα δει. Μεγαλύτερη διαφάνεια σημαίνει βελτιωμένη συνεργασία, η οποία σημαίνει μεγαλύτερη παραγωγικότητα.

- **Security – Ασφάλεια**

Ένα από τα βασικά προβλήματα των επιχειρήσεων είναι οι απώλειες των φορητών υπολογιστών και κατά συνέπεια των δεδομένων τους που κοστίζουν πιο πολύ από τους ίδιους τους υπολογιστές. Με το Cloud computing τα δεδομένα είναι αποθηκευμένα σε ένα σύννεφο - "Cloud " και η πρόσβαση σε αυτά δεν εξαρτάται απ' το τι θα συμβεί στο μηχάνημά. Επιπλέον υπάρχει δυνατότητα εξ αποστάσεως διαγραφή των δεδομένων από τον φορητό υπολογιστή που χάθηκε ώστε να μην πέσουν σε λάθος χέρια.

- **Ανταγωνιστικότητα**

Το Cloud computing γίνεται ανταγωνιστικό προσφέροντας πρόσβαση στην enterprise-class technology στον οποιονδήποτε. Επιτρέπει επίσης σε μικρότερες επιχειρήσεις να δρουν ταχύτερα από ό, τι οι μεγαλύτεροι καθιερωμένοι ανταγωνιστές, διαταράσσοντας την αγορά, ενώ παραμένει λιτή και ευέλικτη. "Ο Δαβίδ που κρύβει μέσα του έναν Γολιάθ".

- **Φιλικό προς το περιβάλλον**

Καθώς τα παραπάνω σημεία περιγράφουν τα οφέλη του Cloud computing για τις επιχειρήσεις και τον χρήστη, δεν μπορούμε να μην αναφερθούμε στην φιλική προς το περιβάλλον επίδραση. Εφόσον στις ανάγκες του Cloud υπάρχουν διακυμάνσεις και οι πόροι δεν χρησιμοποιούνται όταν δεν χρειάζονται, τότε έχουμε χαμηλότερη κατανάλωση ενέργειας, λιγότερες εκπομπές άνθρακα στην ατμόσφαιρα και κατά συνέπεια λιγότερες επιπτώσεις στο περιβάλλον. (Salesforce, n.d.)

1.4 Ακεραιότητα Δεδομένων – Ασφάλεια

Η ακεραιότητα των δεδομένων είναι ένα από τα πιο σημαντικά στοιχεία σε οποιοδήποτε σύστημα πληροφοριών. Η ακεραιότητα των δεδομένων σημαίνει προστασία των δεδομένων από μη εξουσιοδοτημένο χρήστη, που μπορεί να υποκλέψει, να τροποποιήσει, να κατασκευάσει ή να καταστρέψει δεδομένα. Η ακεραιότητα των δεδομένων επιτυγχάνεται εύκολα σε ένα αυτόνομο σύστημα με μια ενιαία βάση δεδομένων όπου συντηρείται μέσω των περιορισμών και τον έλεγχο των κινήσεων στη βάση δεδομένων, η οποία συνήθως γίνεται με ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Οι περισσότερες βάσεις δεδομένων

υποστηρίζουν την διαχείριση αυτή και ακολουθώντας τις ιδιότητες ACID (atomicity, consistency, isolation, και durability) εξασφαλίζεται η ακεραιότητα των δεδομένων.

Για τον έλεγχο της πρόσβασης των δεδομένων χρησιμοποιείται ένας μηχανισμός εξουσιοδότησης με τον οποίο ένα σύστημα καθορίζει το επίπεδο πρόσβασης σε έναν συγκεκριμένο πιστοποιημένο χρήστη που θα πρέπει να εξασφαλίσει τους πόρους που ελέγχονται από το σύστημα.

Η ακεραιότητα των δεδομένων σε ένα σύστημα Cloud σημαίνει διατήρηση της ακεραιότητας των πληροφοριών. Τα δεδομένα δεν πρέπει να χαθούν ή να τροποποιηθούν από μη εξουσιοδοτημένους χρήστες. Η ακεραιότητα των δεδομένων είναι η βάση για την παροχή υπηρεσιών υπολογιστικού νέφους, όπως SaaS, PaaS, και IaaS. Εκτός από την αποθήκευση της μεγάλης κλίμακας δεδομένων, το Cloud computing περιβάλλον παρέχει συνήθως την υπηρεσία επεξεργασίας δεδομένων. Η ακεραιότητα των δεδομένων μπορεί να ληφθεί με τεχνικές όπως RAID και ψηφιακής υπογραφής.

Λόγω της μεγάλης ποσότητας των οντοτήτων και τα σημεία πρόσβασης σε ένα περιβάλλον Cloud, οι άδειες είναι ζωτικής σημασίας για τη διασφάλιση ότι μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να αλληλοεπιδρούν με τα δεδομένα. Με την αποφυγή της μη εξουσιοδοτημένης πρόσβασης, οι οργανισμοί μπορούν να επιτύχουν ακεραιότητα των δεδομένων τους.

Οι μηχανισμοί παρακολούθησης προσφέρουν τη μεγαλύτερη ορατότητα καθορίζοντας ποιος ή τι μπορεί να έχουν αλλάξει στα δεδομένα ή στις πληροφορίες του συστήματος, οι οποίες ενδεχομένως να επηρεάζουν την ακεραιότητα τους.

Οι πάροχοι Cloud computing είναι αξιόπιστοι για να διατηρήσουν την ακρίβεια και την ακεραιότητα των δεδομένων. Ωστόσο, είναι απαραίτητη η κατασκευή ενός τρίτου μηχανισμού εποπτείας - third party tool, που θα επαληθεύει από απόσταση την ακεραιότητα των δεδομένων στο Cloud, παράλληλα με τους χρήστες και τους παρόχους.

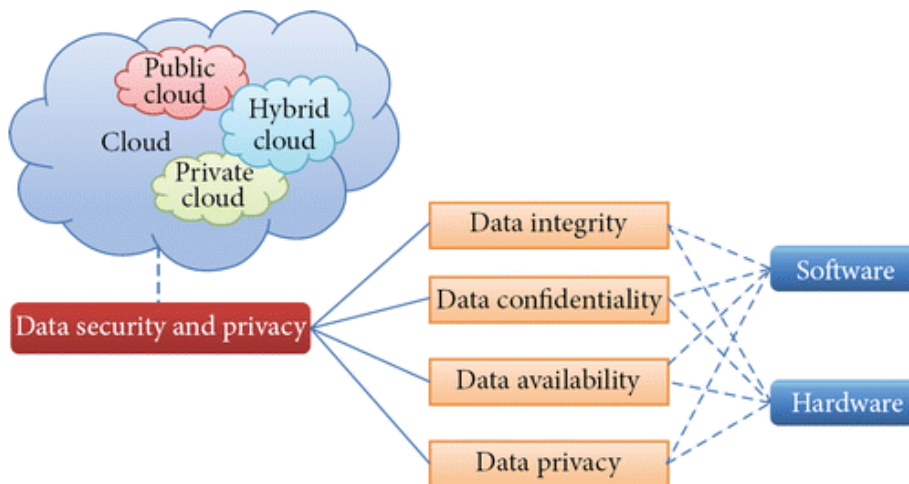


Figure 6: Organization of data security and privacy in cloud computing

1.5 Εμπιστευτικότητα των δεδομένων

Στις μέρες μας, στέλνουμε, λαμβάνουμε, επεξεργαζόμαστε, και αποθηκεύουμε δεδομένα σύμφωνα με τις ανάγκες και τις απαιτήσεις μας. Τα δεδομένα αυτά εμπίπτουν σε κάποιους κανόνες και οι πάροχοι οφείλουν να τους τηρούν. Η επιλογή εγκατάστασης και αποθήκευσης των δεδομένων μας σε ένα Cloud (ιδιωτικό, υβριδικό ή δημόσιο) εξαρτάται από την κατανόηση ότι ο πάροχος είναι πλήρως συμβατός με τα παραπάνω, διαφορετικά, υπάρχει ο κίνδυνος παραβίασης της ιδιωτικότητας και των δεδομένων μας. Οι επιπτώσεις για τη διατήρηση της ασφάλειας των πληροφοριών είναι σημαντικό όταν πρόκειται για την προστασία της ιδιωτικής ζωής.

Υπήρξαν αρκετές παραβιάσεις προσωπικών δεδομένων έξω από τη σφαίρα του Cloud computing για να ξεκινήσει η ανησυχία για οποιοδήποτε σύστημα βασίζεται σε Cloud αρχιτεκτονική ή στον παραδοσιακό τρόπο αποθήκευσης, επεξεργασίας ή μετάδοσης ευαίσθητων πληροφοριών, με δημοφιλέστερο παράδειγμα το 2010, όπου cloud-based υπηρεσίες, συμπεριλαμβανομένων των Facebook, Twitter και Google είχαν διαρροή δεδομένων.

Η προστασία της ιδιωτικής ζωής και οι ανησυχίες στο εσωτερικό του μοντέλου Cloud δεν είναι κάτι νέο. Όπως ο ενοικιαστής με τις νομικές υποχρεώσεις της ιδιωτικότητας και το χειρισμό στα ζητήματα προστασίας της ιδιωτικής ζωής, δεν είναι διαφορετικό, αν χρησιμοποιεί το Cloud. Ακριβώς όπως κανένας χρήστης δεν θα αποθήκευε τα δεδομένα του σε ένα διακομιστή χωρίς επαρκή έλεγχο, έτσι δεν θα επέλεγε έναν οποιοδήποτε πάροχο Cloud χωρίς να επαληθεύσει ότι πληροί τα ίδια κριτήρια αξιολόγησης για το πώς προστατεύει τα δεδομένα σε κατάσταση ηρεμίας, κατά τη μετάδοση ή κατά την επεξεργασία.

Οι πολιτικές προστασίας μπορεί να αποκλείσουν κάθε εξωτερικό πάροχο διαχείρισης ευαίσθητων πληροφοριών του χρήστη, συμπεριλαμβανομένων των παρόχων Cloud. Ενώ απ' την άλλη πλευρά υπάρχει η αντίληψη ότι ο υπολογιστής στο γραφείο είναι ασφαλέστερος από ένα δημόσιο Cloud, είναι πιθανόν να μην είναι, εκτός αν παίρνονται ασυνήθιστες τεχνικές και διαδικαστικές προφύλαξης.

Η ασφάλεια και η διακυβέρνηση είναι δύο ξεχωριστά ζητήματα, και ως μέρος της δέουσας επιμέλειας, θα πρέπει να κατανοήσουμε πλήρως τη διακυβέρνηση της ιδιωτικής ζωής που προσφέρουν οι πάροχοι, καθώς και τις πρακτικές ασφάλειας και τις κατευθυντήριες γραμμές. Όπως με τις προσωπικές πληροφορίες που υπόκεινται στους νόμους περί προστασίας της ιδιωτικής ζωής, διάφορες κατηγορίες επιχειρήσεων πληροφοριών εθνικής ασφάλειας υπόκεινται επίσης σε ρυθμίσεις δικαίου.

Δεδομένου του μεγέθους της κυβέρνησης και ο αριθμός επιπέδων και δικαιοδοσιών, φαίνεται ότι οι ίδιες οι κυβερνήσεις θα μπορούσαν να λειτουργήσουν μια σειρά από κοινοτικά Cloud για την αποκλειστική χρήση τους, λαμβάνοντας έτσι τα οφέλη και αποφεύγοντας τα ζητήματα ασφάλειας με τη συμβίωση σε ένα δημόσιο Cloud. Από την άλλη πλευρά, αν οι κυβερνήσεις ήταν να χρησιμοποιήσουν δημόσια Cloud, τότε οι υπηρεσίες αυτές θα έπρεπε να πληρούν τα συμφέροντα των μισθωτών και όλους τους ισχύοντες κανονισμούς και νόμους.

Είναι πιθανό ο μισθωτής να έπρεπε να εφαρμόσουν πρόσθετους ελέγχους ασφαλείας που πληρούν τις ρυθμιστικές ή νομικές απαιτήσεις ακόμα και όταν η υποκείμενη δημόσια "υποδομή ως υπηρεσία (IaaS)" ή "Πλατφόρμα ως υπηρεσία (PaaS)" δεν ανταποκρίνονται πλήρως στις ίδιες αυτές απαιτήσεις. Ωστόσο, πρέπει να γίνει κατανοητό ότι το εύρος των πρόσθετων ελέγχων που μπορούν να προστεθούν από ένα μισθωτή είναι περιορισμένο και δεν μπορούν να ξεπεράσουν πολλά εμπόδια σε ορισμένες δημόσιες υπηρεσίες Cloud. (Winkler, 2011)

Η εμπιστευτικότητα των δεδομένων είναι σημαντικό για τους χρήστες που αποθηκεύουν προσωπικά ή εμπιστευτικά δεδομένα στο Cloud. Οι στρατηγικές ελέγχου ταυτότητας και ελέγχου πρόσβασης χρησιμοποιούνται για τη διασφάλιση των εν λόγω δεδομένων.

Επειδή οι χρήστες δεν εμπιστεύονται τους παρόχους Cloud και οι πάροχοι υπηρεσιών Cloud αδυνατούν να εξαλείψουν πιθανή απειλή, είναι πολύ επικίνδυνο για τους χρήστες να αποθηκεύουν τα ευαίσθητα δεδομένα τους στο σύννεφο. (Yunchuan Sun, 2014)

1.5.1 Κίνδυνοι στο σύννεφο

Η παροχή της υπηρεσίας του Cloud εκμεταλλεύεται την υποδομή του παρόχου. Ο χρήστης εκχωρεί μεγάλο μέρος του ελέγχου στον πάροχο και ταυτόχρονα η συμφωνία στο κομμάτι των υπηρεσιών ανάμεσα σε πελάτη και πάροχο (SLA) δεν αποσαφηνίζει αυτό το θέμα, αφήνοντας ερωτηματικά και κενά σε επίπεδο ασφάλειας. Τι θα συνέβαινε στην πραγματικότητα αν χαθεί ο έλεγχος από τον πάροχο;

Στις μέρες μας τα εργαλεία που προσφέρουν οι πάροχοι, οι διαδικασίες, το πρότυπο τυποποιημένης μορφής και οι υπηρεσίες που έχουν την δυνατότητα να παρέχουν ασφάλεια δεδομένων, και η φορητότητα σε άλλο πάροχο, δεν μπορούμε να θεωρήσουμε ότι έχουν φτάσει σε υψηλό επίπεδο. Με άλλα λόγια, αν θελήσει ο χρήστης ή μια επιχείρηση να αλλάξει πάροχο υπηρεσιών Cloud έχει δύο προβλήματα. Το πρώτο είναι η δυσκολία της μεταφορά αρχείων και υπηρεσιών, ενώ θα γινόταν δυσκολότερο αν θελήσει να γυρίσει στην κλασική λύση με clients και server εντός της επιχείρησης. Η φορητότητα των δεδομένων δεν είναι πάντα εφικτή και δεν γίνεται να γυρίσουμε πίσω στον παλιό κλασικό server σε όλες τις περιπτώσεις. Σε αυτό το σημείο βέβαια, τίθεται θέμα εξάρτησης από τον πάροχο.

Η κοινή διαχείριση αρχείων και πόρων που προσφέρεται από το Cloud έχει ένα μειονέκτημα συγκριτικά με τη multi-tenant αρχιτεκτονική. Για να κατανοήσουμε τον όρο multi-tenant, «φανταστείτε το δίκτυο σαν ένα κτήριο. Όλοι μπορούν να χρησιμοποιήσουν τον ανελκυστήρα αλλά στο μηχανοστάσιο πρόσβαση έχει μόνο ο διαχειριστής. Το πρόβλημα σε αυτή την περίπτωση είναι πώς μπορεί να διασφαλιστεί το δίκτυο από τους ενοίκους, στο σύνολο τους αλλά και ατομικά. Αν για παράδειγμα, ένα διαμέρισμα του 3ου ορόφου επικοινωνεί με εσωτερική πόρτα με το διπλανό του διαμέρισμα τότε αν ένας κλέφτης προσπαθήσει να μπει στο πρώτο αλλά συναντήσει μια πόρτα ασφαλείας αδιαπέραστη και θωρακισμένη τότε πάει στο διπλανό που επικοινωνούν, στο οποίο η πόρτα είναι ανοιχτή, μπαίνει εύκολα και το χειρότερο είναι ότι αποκτά πρόσβαση και στο πρώτο διαμέρισμα». Θεωρητικά, έτσι θα μπορούσε να επιτευχθεί μία επίθεση τύπου guest-hopping. Σκεφτείτε τις εικονικές μηχανές σαν διαμερίσματα και τον επόπτη τους (virtual machine monitor ή hypervisor). Στην πραγματικότητα όμως, ο βαθμός δυσκολίας μιας τέτοιας επίθεσης είναι υψηλός σε σχέση με τις κλασικές επιθέσεις στα παραδοσιακά λειτουργικά.

Για το λόγο του ότι η ασφάλεια δεν είναι μόνο το τεχνικό κομμάτι και ο κώδικας ενός υπολογιστή, ένας κίνδυνος που αναφέρεται στις επιχειρήσεις είναι και ο παρακάτω. Σε μια επένδυση που πρέπει να συμμορφωθεί σε συγκεκριμένους όρους πιθανόν να μην είναι καλή ιδέα να εμπλακεί στην υπηρεσία του Cloud, πόσο μάλλον στην περίπτωση όπου ο πάροχος δεν δίνει κάποιο αποδεικτικό ότι η υπηρεσία που προσφέρει θα δουλεύει σύμφωνα με τις απαιτήσεις. Πρέπει να προσέξουμε λοιπόν θέματα που αναφέρονται σε κανονιστικά πλαίσια και πρότυπα λειτουργίας.

Το σύννεφο μας δίνει την δυνατότητα να έχουμε πρόσβαση από παντού μέσω παγκόσμιου ιστού και κοινόχρηστων δικτύων. Είναι κατανοητό πως υπάρχει μεγάλος κίνδυνος στην ασφάλεια, ειδικότερα όταν συνδυάζεται και με απομακρυσμένη σύνδεση.

Πόσο σίγουροι μπορούμε να νιώθουμε για την ασφάλεια των δεδομένων μας; Ποιος μας εγγυάται ότι δεν θα χαθεί κάποιο σημαντικό αρχείο; Γνωρίζουμε ότι τα αρχεία μας στο σύννεφο μετακινούνται μεταξύ δικτύων ανά τον κόσμο και βασική ερώτηση είναι, ελέγχονται από κάποιον; Ποιος είναι αυτός; Και, αν κάποιος τα ελέγχει, ελέγχει και πού πάνε; Τα διαχειρίζεται με νόμιμο τρόπο; Γενικά, είναι αναγκαίο να γνωρίζουμε την πολιτική του παρόχου που χρησιμοποιούμε. Κάποιοι μάλιστα παρέχουν και πιστοποιητικά.

Όσον αφορά το θέμα των αρχείων, ένας κίνδυνος παρουσιάζεται κατά τη διαγραφή τους. Είναι γνωστό βέβαια πως αν θέλαμε να σβήσουμε τα πάντα από το δίσκο υπάρχουν πολλές τεχνικές. Από τις πιο σύνθετες που καθιστούν δύσκολο έως αδύνατο την ανάκτηση τους, μέχρι ένα απλό delete. Σε ένα Cloud όμως, επειδή μοιραζόμαστε το δίσκο και με άλλους χρήστες δεν θα μπορούσαμε να κάνουμε κάτι τέτοιο γιατί μην ξεχνάμε το γεγονός ότι το σβήσιμο δεν γίνεται πάντα σε πραγματικό χρόνο.

Υποθέτοντας πάντα ότι ένα Cloud είναι εξωτερικά διασφαλισμένο, τι θα γινόταν στην περίπτωση που η απειλή ξεκινάει από την υποδομή; Με ποιον τρόπο διασφαλιζόμαστε εμείς; Οι κίνδυνοι που αναφέρονται παραπάνω θεωρούνται οι σημαντικότεροι και πάνω σε αυτούς θα χτίσουμε και την άλλη πλευρά του νομίσματος. Ας δούμε τώρα τι προσφέρει το Cloud και ποια είναι τα πλεονεκτήματα του στην ασφάλειά.



Figure 7: Cloud computing security

1.5.2 Πλεονεκτήματα Ασφαλείας στο Cloud computing

Οι προοπτικές του Cloud ως τεχνολογία για να αναδείξει την ασφάλεια του σε όλους τους τομείς είναι μεγάλες. Παρακάτω θα δούμε τα πλεονεκτήματα ασφαλείας στο Cloud computing.

- Τα μέτρα ασφαλείας έχουν λιγότερο κόστος γιατί φτιάχνονται για δίκτυα ευρείας κλίμακας
- Οι περισσότεροι πάροχοι μεταφέρουν τα δεδομένα σε πολλές περιοχές ώστε να εξασφαλίσουν ότι τα δεδομένα είναι πάντα διαθέσιμα
- Η ταχύτητα λήψης των δεδομένων και η αποθήκευσή τους σε τοπικό επίπεδο παρέχονται με την βέλτιστη απόδοση
- Εξειδικευμένο προσωπικό για τον κάθε ένα ξεχωριστά τομέα ασφαλείας αντίθετα με τις μικρομεσαίες εταιρείες που συνήθως όλοι ασχολούνται με όλα

- Κωδικοποίηση και αύξηση των μέτρων ασφάλειας ώστε να αποφευχθούν οι απειλές
- Οι πελάτες χρησιμοποιούν virtual μηχανές που έχουν τις τελευταίες ενημερώσεις για ιούς και κενά ασφάλειας

Σύμφωνα με τα παραπάνω είναι κατανοητό ότι οι πάροχοι υπηρεσίας Cloud έχουν την δυνατότητα να προσφέρουν καλύτερες, πιο εξελιγμένες και με χαμηλότερο κόστος υπηρεσίες ασφάλειας στους πελάτες τους. Ταυτόχρονα το ίδιο συμβαίνει και με τους πόρους του δικτύου. (A.Souhalakis, 2010)

1.5.3 Μέτρα προστασίας

Ενώ το κόστος της ενσάρκωσης virtual security συσκευών στο Cloud είναι χαμηλότερο, η τεχνολογία είναι πιο καινούργια και οι προβληματισμοί για την προστασία των δεδομένων μας στο σε ένα Cloud είναι αρκετοί. Παρακάτω θα δούμε κάποια βασικά μέτρα προστασίας που μπορούμε να πάρουμε έτσι ώστε να είμαστε όσο το δυνατόν πιο ασφαλείς.

- Αρχικά πρέπει να τονιστεί ότι χρειάζεται ιδιαίτερη προσοχή στο τι ανοίγουμε. Ένα κακόβουλο μήνυμα ή ένας σύνδεσμος φτάνουν για να γίνει υποκλοπή των δεδομένων μας
- Ενημερωνόμαστε πάντα για τις παροχές που μας προσφέρονται από έναν πάροχο και δεν ξεχνάμε ποτέ να κάνουμε backup, κάποιοι πάροχοι μας δίνουν αυτή την υπηρεσία
- Βασική αδυναμία του Cloud είναι ότι δεν υπάρχει καλή κρυπτογράφηση των δεδομένων ώστε να μας δώσει περισσότερη ασφάλεια στα δίκτυα, αυτό το μπορεί να το καταστήσει εύκολο σε επιθέσεις

Τις περισσότερες φορές η χρήση του Cloud γίνεται μέσω ενός υπολογιστή από έναν μέσο χρήστη μη διαθέτοντας κάποιο σύστημα ασφάλειας. Αυτό επιφέρει την αδυναμία να αντιληφθεί πολλές φορές κακόβουλα στοιχεία που εισχωρούν στα δεδομένα του. Γι' αυτό τον λόγο οι πάροχοι που προσφέρουν τις υπηρεσίες αυτές πρέπει να προβλέπουν και να μην περιμένουν να δράσουν αφού δεχτούν επίθεση εν αγνοία του χρήστη. (A.Souhalakis, 2010)

1.6 Cloud computing, Private, Public ή και τα δύο;

Ένας αυξανόμενος αριθμός οργανισμών σε όλο τον κόσμο βλέπουν την αξία του Cloud computing ως έναν τρόπο αύξησης της ευελιξίας της τεχνολογίας πληροφορικής και της in-house μείωσης του κόστους των υποδομών. Ωστόσο, ο σωστός συνδυασμός ασφάλειας, ελέγχου και κόστους εξαρτάται από την επιλογή της σωστής δημόσιας ή ιδιωτικής υποδομής (Public or Private Infrastructure)- ή και τα δύο.

Το Cloud computing υπερέρχει σαν λύση, με πολλούς μεγάλους οργανισμούς να το προτιμούν, όπως η Volkswagen AG, μια από τις μεγαλύτερες κατασκευάστριες εταιρίες αυτοκινήτων που χρησιμοποιεί το Cloud OpenStack για την διανομή του ιστοτόπου της. Επίσης το CERN (European Organization for Nuclear Research) χρησιμοποιεί το OpenStack Cloud για να προσφέρει τους απαιτούμενους υπολογιστικούς πόρους για να αναλύουν και να μοντελοποιούν δεδομένα από τον Large Hadron Collider (LHC).

Η μεγάλη στροφή προς το Cloud αποφέρει κέρδη κόστους και αποδοτικότητας για επιχειρήσεις σε όλους τους τομείς, εξαιτίας των μειωμένων απαιτήσεων και το κόστος των υποδομών, έως

την κλιμακούμενη πρόσβαση σε δίκτυα, τους υπολογιστικούς πόρους και την αποθήκευση. (Kyle, 2012)

1.6.1 Private Cloud, πλεονεκτήματα και μειονεκτήματα

Ορισμένοι οργανισμοί, όπως Χρηματοπιστωτικά ιδρύματα και η ιατρική βιομηχανία, έχουν στην κατοχή τους ευαίσθητα δεδομένα συναλλαγών και πελατών τα οποία πρέπει να αποθηκεύονται σε διακομιστές εντός του εταιρικού τείχους προστασίας. Αν λοιπόν αυτό συμβαίνει σε αυτές τις επιχειρήσεις, ένα ιδιωτικό σύννεφο (Private Cloud) είναι μονόδρομος ώστε να μην υπάρξει διαρροή προσωπικών δεδομένων. Κατ' επέκταση και πολλοί άλλοι οργανισμοί θεωρούν πιο ασφαλές τα δεδομένα τους να βρίσκονται σε ιδιόκτητο Cloud.

Ενώ τα οφέλη από τα ιδιωτικά σύννεφα είναι σαφή, υπάρχουν και κάποια μειονεκτήματα που πρέπει να ληφθούν υπόψη. Για παράδειγμα, θα υπάρχει πάντοτε ένα ανώτατο όριο χωρητικότητας λόγω των περιορισμών του φυσικού υλικού στο δωμάτιο διακομιστή του οργανισμού. Επιπλέον, πολλοί ιδιωτικοί προμηθευτές Cloud "κλειδώνουν" τους πελάτες με κάποιο ιδιόκτητο λογισμικό, API και πρότυπα - αυξάνοντας το κόστος. (Kyle, 2012)

1.6.2 Public Cloud, πλεονεκτήματα και μειονεκτήματα

Σε ένα δημόσιο σύννεφο (Public Cloud) δεν υπάρχουν όρια στη χωρητικότητα. Πάροχοι όπως Amazon, Rackspace, Dell και HP, παρέχουν στις επιχειρήσεις άμεση και οικονομική αποδοτική πρόσβαση σε ηλεκτρονικούς υπολογιστές και πόρους αποθήκευσης on demand, χωρίς να υπάρχει ανάγκη για υποδομή στο χώρο. Τα δημόσια σύννεφα μειώνουν δραματικά τις απαιτήσεις υλικού, διαχείρισης και κατά συνέπεια κόστους. Επιπλέον, υποστηρίζουν την ταχεία ανάπτυξη νέων υπηρεσιών και επιτρέπουν την κλιμάκωση.

Τα δημόσια σύννεφα είναι ιδανικά κατάλληλα για τη διαχείριση μεγάλου όγκου εργασιών ρουτίνας που παρουσιάζουν ελάχιστους κινδύνους ασφαλείας. Προσφέρουν επίσης μια εξαιρετική λύση όπου απαιτούνται υπολογιστικοί πόροι σε προσωρινή βάση, για παράδειγμα, κατά την ανάπτυξη και δοκιμή νέων εφαρμογών.

Για να παρέχουν πρόσθετη ευελιξία, τα δημόσια νέφη επιτρέπουν στους πελάτες να επιλέγουν το προτιμώμενο λειτουργικό τους σύστημα, μειώνοντας τον κίνδυνο κλειδώματος από τον πωλητή όπως σε ένα Private Cloud και τους περιορισμούς στη λειτουργικότητα τους.

Από την άλλη πλευρά, τα δημόσια σύννεφα είναι περιβάλλοντα πολλαπλών μισθώσεων που θεωρούνται λιγότερο ασφαλή από τα ιδιωτικά σύννεφα. Είναι σίγουρα αλήθεια ότι οι πελάτες δεν έχουν κανέναν έλεγχο της υποκείμενης υποδομής, τοποθετώντας όλα τα δεδομένα τους εκεί.

Κάθε δημόσιος φορέας παροχής υπηρεσιών Cloud έχει τα δικά του SLA για ασφάλεια, δημιουργία αντιγράφων ασφαλείας, αποκατάσταση καταστροφών και διαθεσιμότητα, γι' αυτό το λόγο πρέπει να ελεγχθούν όλα αυτά από τους υποψήφιους πελάτες πριν προχωρήσουν στην λύση αυτή. (Kyle, 2012)

1.7 Hybrid cloud

Hybrid Cloud ονομάζουμε τον συνδυασμό ενός Private και ενός Public cloud επιτρέποντας την ανταλλαγή δεδομένων και εφαρμογών μεταξύ τους.

Η λύση του Hybrid cloud computing δίνει την δυνατότητα στις επιχειρήσεις να κλιμακώνουν ανάλογα την υποδομή τους εντός του Private cloud μέχρι το Public cloud έτσι να μπορούν να

χειριστούν οποιαδήποτε υπερχείλιση (overflow) χωρίς να παρέχουν πρόσβαση σε τρίτους (third-party datacenters) στα δεδομένα τους.

Κέρδος από το συγκεκριμένο περιβάλλον είναι η ευελιξία της υπολογιστικής ισχύς που παρέχεται από το Public cloud για βασικά και μη ευαίσθητα δεδομένα, διατηρώντας ταυτόχρονα εφαρμογές και δεδομένα κρίσιμης σημασίας για την επιχείρηση ασφαλή πίσω από το firewall. (Microsoft, n.d.)

1.8 Οι μεγαλύτεροι πάροχοι Cloud computing

▪ Amazon Web Services

Με μερίδιο αγοράς 31% η Amazon Web Services (AWS), θυγατρική της Amazon.com Inc ξεκίνησε το 2006 και σήμερα έχει εξελιχθεί σε κορυφαίο παίκτη της παγκόσμιας αγοράς του Cloud computing.

Η AWS προσφέρει ένα ευρύ φάσμα υπηρεσιών Cloud computing στους πελάτες της σε όλο τον κόσμο. Αν και το κυριότερο προϊόν του είναι το Elastic Compute Cloud (EC2), υπάρχουν σήμερα περισσότερες από 70 επιλογές της AWS που θα μπορούσαν να χρησιμοποιηθούν για storage, networking, computing, database, και analytics.

Ως νέος παίκτης στον κλάδο, η Amazon σύντομα έγινε ο πάροχος υπηρεσίας παροχής υπηρεσιών για τις νεοσύστατες επιχειρήσεις, οι οποίες σήμερα αποτελούν περίπου τα δύο τρίτα του 1 εκατομμυρίου των πελατών της. Οι πωλήσεις της κατά το δεύτερο τρίμηνο του 2016 ήταν τριπλάσιες σε σχέση με την Microsoft, του πλησιέστερου ανταγωνιστή της. Ως εκ τούτου, ο ηγετικός ρόλος της στον κλάδο δεν αμφισβητείται επί του παρόντος. (Olanubi, 2012)

▪ Microsoft

Η Microsoft ιδρύθηκε το 1974 και με μερίδιο αγοράς στο Cloud computing να είναι στο 11% προσφέρει στους πελάτες της καινοτόμες υπηρεσίες μέσω πλατφόρμων όπως τα Azure, Office 365 και Dynamics 365 από 30 datacenters σε όλο τον κόσμο.

Το πελατολόγιο του Microsoft cloud είναι ποικίλο, με Δημόσιες υπηρεσίες, εταιρικές οργανώσεις και ιδιώτες να υιοθετούν την υποδομή αυτή για να αυτοματοποιήσουν τις δραστηριότητές τους.

Σε μια κίνηση για την αύξηση του μεριδίου της στην αγορά και την επέκταση των δραστηριοτήτων της, η Microsoft πρόσφατα αύξησε την επένδυση σε Cloud computing στην Ευρώπη στα 3 δισεκατομμύρια δολάρια. (Olanubi, 2012)

▪ IBM

Η IBM με μερίδιο αγοράς 8% κατατάσσεται σήμερα ως παγκόσμιος ηγέτης στον τομέα των υβριδικών και ιδιωτικών Cloud μέσω των καινοτόμων υπηρεσιών της που παρέχουν ασφαλή ενσωμάτωση μεταξύ δημοσίων και ιδιωτικών περιβαλλόντων Cloud. Ως εκ τούτου, η ανάπτυξη της επιχείρησης εξαρτάται σε μεγάλο βαθμό από τα έσοδα από τις παγκόσμιες υβριδικές και ιδιωτικές τεχνολογίες Cloud που αναπτύσσονται σε περισσότερα από 40 datacenters ανά τον κόσμο.

Εκτιμάται ότι πάνω από το 80% των επιχειρήσεων που εκμεταλλεύονται λύσεις πληροφορικής θα υιοθετήσουν υβριδικές υπηρεσίες Cloud έως το 2017. Οι υποδομές υβριδικών σύννεφων έχουν αυξήσει δημοτικότητα τους, επειδή βοηθούν τους οργανισμούς σε σημαντικές

εξοικονομήσεις κόστους και βελτίωση της αποδοτικότητας. Ως αποτέλεσμα, παρόλο που ελέγχει σήμερα μόνο το 8% της αγοράς σύννεφων, η IBM είναι έτοιμη να αμφισβητήσει την κυριαρχία των δύο κορυφαίων παικτών οι οποίοι ισχυρίζονται ότι κατέχουν μερίδιο αγοράς 42%. (Olanubi, 2012)

▪ Google

Η Google με μερίδιο της παγκόσμιας αγοράς της τάξης του 4% έχει παρακολουθήσει εντυπωσιακή αύξηση των εσόδων και των κερδών της τα τελευταία χρόνια. Ωστόσο, αυτό οφείλεται κυρίως στα έσοδα από τη διαφήμιση, ενώ τα έσοδα από τις υπηρεσίες Cloud εξακολουθούν να μένουν πίσω, παρά το γεγονός ότι δείχνουν σημάδια ανόδου. Το Google Cloud παρουσίασε έσοδα ύψους 896 εκατομμυρίων δολαρίων κατά το τέταρτο τρίμηνο του 2015, ποσοστό που αντιπροσωπεύει αύξηση 65% από το προηγούμενο τρίμηνο και τα έσοδά του για το 2016 προβλέπεται να φτάσουν τα 4 δισεκατομμύρια δολάρια. Η Google είναι επίσης σήμερα η ταχύτερα αναπτυσσόμενη πλατφόρμα στον κλάδο του Cloud computing που αυξήθηκε κατά 162% το 2ο τρίμηνο του 2016, περισσότερο από το διπλάσιο της Amazon που καταλαμβάνει την πρώτη θέση, η οποία σημείωσε αύξηση 53%.

Η εταιρεία διαθέτει επίσης 1 εκατομμύριο ενεργούς χρήστες στην υπηρεσία Cloud "Google Drive for Work". Αν και κατέχει σήμερα την τέταρτη θέση όσον αφορά το συνολικό μερίδιο αγοράς, η Google κατέχει την τρίτη θέση όσον αφορά τις μεγαλύτερες πλατφόρμες υποδομής ως υπηρεσία (IaaS) και με άλλες καινοτόμες υπηρεσίες που προσφέρονται, η Google θα μπορούσε να δει αύξηση της πελατείας της στο εγγύς μέλλον.

Σε μια πρόσφατη κίνηση για να αυξήσει το μερίδιο στην αγορά του Cloud, η Google προσέλαβε τον συνιδρυτή της VMware, Diane Greene, να ηγηθεί της επιχείρησης υπηρεσιών Cloud της εταιρείας και αναμένεται ότι αυτή η κίνηση θα αυξήσει το ανταγωνιστικό της πλεονέκτημα στον κλάδο σύντομα. (Olanubi, 2012)

▪ Salesforce

Από τότε που ξεκίνησε την καινοτόμο υπηρεσία Cloud Customer Management Management (CRM) γνωστή ως Cloud Sales το 1999, η Salesforce εξελίχθηκε για να γίνει σημαντικός παίκτης στην αγορά του Cloud computing. Με την πάροδο των χρόνων, προστέθηκαν άλλες υπηρεσίες όπως το Service Cloud, το Marketing Cloud και το IoT Cloud στις καινοτόμες υπηρεσίες Cloud που προσφέρθηκαν από την εταιρεία. Με την απρόσκοπτη και ασφαλή αρχιτεκτονική του λογισμικού, η υπηρεσία Cloud CRM βοηθά τόσο τους μικρούς όσο και τους μεγάλους οργανισμούς να διαχειρίζονται τα δεδομένα τους από οποιαδήποτε τοποθεσία.

Ωστόσο, η Salesforce δεν αποτελεί σημαντική πρόκληση για την κυριαρχία των κορυφαίων ανταγωνιστών της, καθώς η αύξηση των εσόδων της έχει υστερήσει τα τελευταία χρόνια. Ως εκ τούτου, κατέχει την 5η θέση στον κλάδο με μερίδιο αγοράς 4%. (Olanubi, 2012)

2. Hypervisors

Hypervisors είναι virtual machine monitor (VMM) που επιτρέπει την ταυτόχρονη εκτέλεση πολλών εικονικών λειτουργικών συστημάτων (virtual operating systems) σε ένα σύστημα υπολογιστή. Αυτές οι εικονικές μηχανές αναφέρονται επίσης ως guest machines και όλοι μοιράζονται το υλικό της φυσικής μηχανής όπως μνήμη, επεξεργαστή, αποθήκευση (Storage) και άλλους σχετικούς πόρους.

Ο μηχανισμός εξακρίβωσης (hypervisor isolates) απομονώνει τα λειτουργικά συστήματα από την κύρια κεντρική μηχανή.

Η δουλειά ενός hypervisor είναι να ανταποκρίνεται στις ανάγκες ενός λειτουργικού συστήματος επισκεπτών και να το διαχειρίζεται αποτελεσματικά. Κάθε εικονική μηχανή είναι ανεξάρτητη και δεν παρεμβαίνει στις υπόλοιπες, αν και τρέχουν στην ίδια μηχανή υποδοχής. Ακόμη και αν κάποια εικονικά μηχανήματα καταρρεύσουν ή αντιμετωπίσουν προβλήματα, τα υπόλοιπα μηχανήματα συνεχίζουν να εκτελούνται κανονικά. (cloudoye, n.d.)

2.1 Τύποι Hypervisor

Οι Hypervisors χωρίζονται σε δύο τύπους.

Ο ένας τύπος είναι ο bare-metal hypervisor που αναπτύσσεται κατευθείαν στο υλικό του συστήματος του κεντρικού υπολογιστή χωρίς υποκείμενα λειτουργικά συστήματα ή λογισμικό. Μερικά παραδείγματα των hypervisors αυτού του τύπου είναι ο hypervisor της Microsoft Hyper-V, το VMware ESXi, το Citrix XenServer και ο KVM. (cloudoye, n.d.)

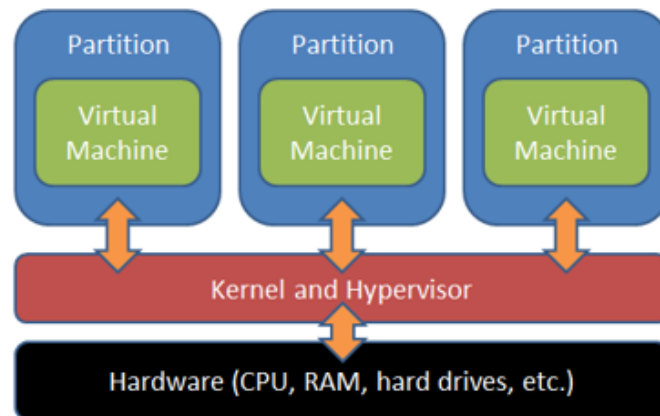


Figure 8: Type 1 Hypervisor

Ο τύπος δύο είναι ένας φιλοξενούμενος (hosted-hypervisor) που τρέχει ως software layer μέσα σε ένα φυσικό λειτουργικό σύστημα. Ο hypervisor τρέχει ως ξεχωριστό δεύτερο στρώμα πάνω από το hardware, ενώ το λειτουργικό σύστημα λειτουργεί ως τρίτο στρώμα. Στους φιλοξενούμενους hypervisors περιλαμβάνονται τα Parallels Desktop και VMware Player. (cloudoye, n.d.)

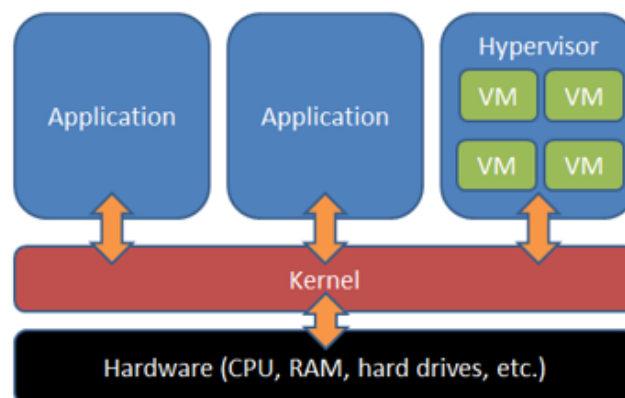


Figure 9: Type 2 Hypervisor

3. OpenStack

Το OpenStack είναι ένα σύνολο εργαλείων λογισμικού για την κατασκευή και διαχείριση πλατφόρμων Cloud computing για δημόσια και ιδιωτικά Clouds. Υποστηριζόμενο από μερικές από τις μεγαλύτερες εταιρείες ανάπτυξης λογισμικού και φιλοξενίας (hosting), καθώς και από χιλιάδες μεμονωμένα μέλη. Πολλοί πιστεύουν ότι το OpenStack είναι το μέλλον του Cloud computing. Το OpenStack είναι υπό την διαχείριση του OpenStack Foundation, ενός μη κερδοσκοπικού οργανισμού που επιβλέπει τόσο την ανάπτυξη όσο και το community-building γύρω από το έργο. (Openstack, n.d.)

3.1 Εισαγωγή στο OpenStack

Το OpenStack επιτρέπει στους χρήστες να αναπτύξουν εικονικές μηχανές και άλλα instances που χειρίζονται διαφορετικές εργασίες για τη διαχείριση ενός περιβάλλοντος Cloud σε περιόδους λειτουργίας. Κάνει την επεκτασιμότητα ευκολότερη, πράγμα που σημαίνει ότι μπορούν εύκολα να εξυπηρετηθούν περισσότεροι χρήστες ταυτόχρονα ανάλογα την περίπτωση. Το πιο σημαντικό όμως είναι ότι το OpenStack είναι λογισμικό ανοιχτού κώδικα, το οποίο σημαίνει ότι όποιος επιλέγει να έχει πρόσβαση στον πηγαίο κώδικα, μπορεί να κάνει οποιοσδήποτε αλλαγές ή τροποποιήσεις που χρειάζεται και να τις μοιραστεί ελεύθερα με την κοινότητα. Αυτό πρακτικά σημαίνει ότι το OpenStack έχει το πλεονέκτημα να δουλεύουν χιλιάδες προγραμματιστές μαζί σε όλο τον κόσμο και έχει σαν αποτέλεσμα να το αναπτύξουν ισχυρότερο, πιο ανθεκτικό και πιο ασφαλές προϊόν που μπορούν. (Openstack, n.d.)



Figure 10 : OpenStack Cloud Software

3.1.1 OpenStack Foundation

Το OpenStack Foundation προωθεί την παγκόσμια ανάπτυξη, διανομή και υιοθέτηση του λειτουργικού συστήματος Cloud OpenStack. Ως παγκόσμια ανεξάρτητη στέγη για το OpenStack, εξυπηρετεί περισσότερα από 60.000 μεμονωμένα μέλη από περισσότερες από 180 χώρες σε όλο τον κόσμο.

Ο στόχος του OpenStack Foundation είναι να εξυπηρετεί τους προγραμματιστές, τους χρήστες και ολόκληρο το οικοσύστημα σε παγκόσμιο επίπεδο, παρέχοντας ένα σύνολο από κοινούς πόρους για να αυξήσει το αποτύπωμα των δημόσιων και ιδιωτικών Clouds OpenStack, και να βοηθήσει τους προγραμματιστές να δημιουργήσουν το καλύτερο Cloud λογισμικού.

Όπως και το λογισμικό, η συμμετοχή στο OpenStack Foundation είναι δωρεάν και προσβάσιμη σε οποιονδήποτε. Τα μεμονωμένα μέλη αναμένεται να συμμετάσχουν στην κοινότητα OpenStack μέσω τεχνικών συμβουλών και οικοδόμησης κοινότητας αυτής. (Openstack foundation, n.d.)

3.2 Αρχιτεκτονική

Η αρχιτεκτονική της εφαρμογής που ακολουθεί όπως βλέπουμε και στο διάγραμμα παρακάτω απαιτεί τουλάχιστον δύο κόμβους (κεντρικούς υπολογιστές) να εκκινήσουν μια βασική εικονική μηχανή (VM) ή ένα instance. Οι προαιρετικές υπηρεσίες όπως το Block Storage και το Object Storage απαιτούν επιπλέον κόμβους.

Αυτό το παράδειγμα αρχιτεκτονικής διαφέρει από μια ελάχιστη αρχιτεκτονική παραγωγής ως εξής:

Οι Networking agents βρίσκονται στον κόμβο του Controller αντί ενός ή περισσότερων εξειδικευμένων κόμβων δικτύου (Network nodes) όπως συμβαίνει για παράδειγμα στην προγενέστερη έκδοση Juno.

Το Overlay (tunnel) traffic για δίκτυα αυτοεξυπηρέτησης (self-service networks) διασχίζει το δίκτυο διαχείρισης (Management network) αντί για ένα αποκλειστικό δίκτυο. (OpenStack, 2017)

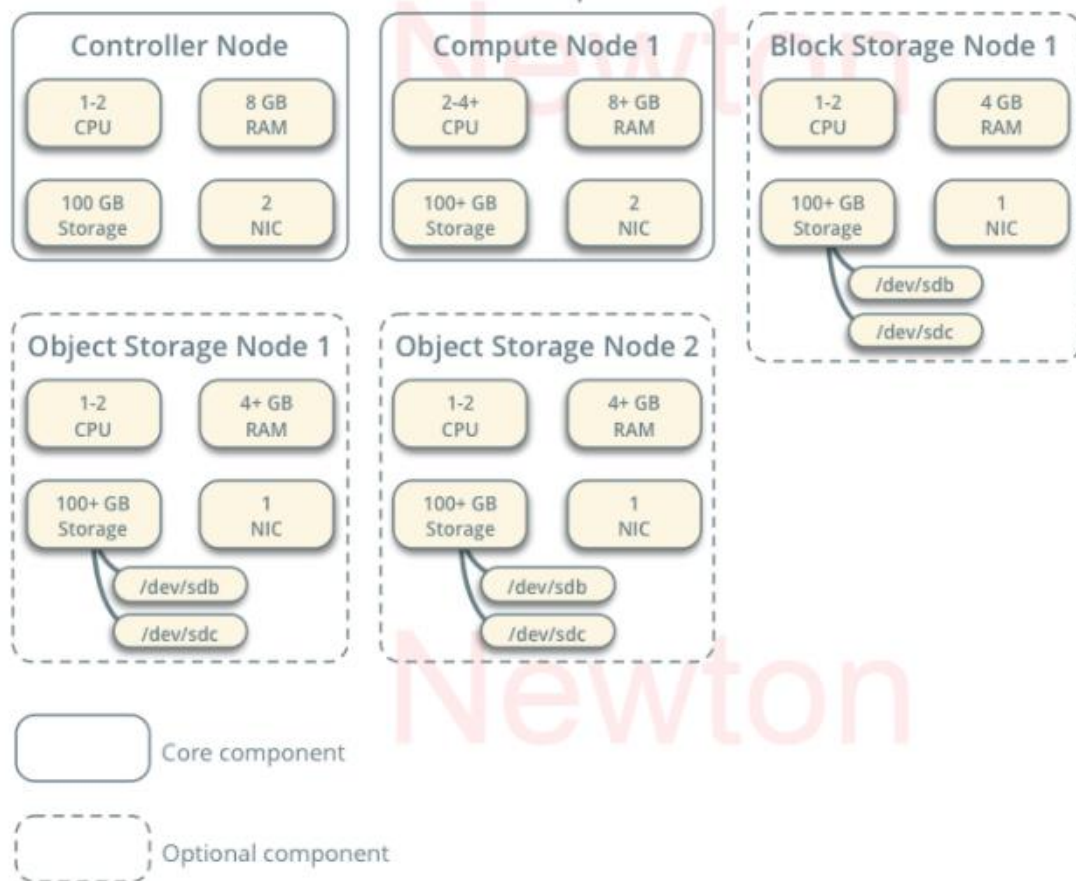


Figure 11 : OpenStack Newton hardware requirements

3.2.1 Controller

Ο Controller node απαιτεί τουλάχιστον δύο διεπαφές δικτύου (network interfaces) και εκτελεί την υπηρεσία ταυτότητας (Identity service), το Image service, τα τμήματα διαχείρισης του Compute (management portions), το management networking, διάφορους Networking agents και τον πίνακα ελέγχου (Dashboard). Περιλαμβάνει επίσης υπηρεσίες υποστήριξης, όπως την βάση δεδομένων SQL, το message queue και το NTP.

Προαιρετικά, ο κόμβος του ελεγκτή εκτελεί τμήματα των υπηρεσιών Block Storage, Object Storage, Orchestration και Telemetry. (OpenStack, 2017)

3.2.2 Compute

Ο Compute node απαιτεί και αυτός τουλάχιστον δύο διεπαφές δικτύου και τρέχει τον hypervisor του Compute που χρησιμοποιούν τα instances ο οποίος από προεπιλογή είναι ο KVM hypervisor. Ο Compute node τρέχει επίσης έναν Networking service agent που συνδέει τα instances με εικονικά δίκτυα και παρέχει υπηρεσίες firewall σε αυτά μέσω των security groups.

Μπορούν να αναπτυχθούν περισσότεροι από έναν Compute node. (OpenStack, 2017)

3.2.3 Block Storage

Ο προαιρετικός Block Storage node απαιτεί τουλάχιστον μία διασύνδεση δικτύου και περιέχει τους δίσκους που παρέχουν οι υπηρεσίες Block Storage και Shared File System για τα instances.

Για την απλή λειτουργία, η κυκλοφορία υπηρεσιών (service traffic) μεταξύ των Compute nodes και αυτού του κόμβου χρησιμοποιείτε το δίκτυο διαχείρισης (Management network). Τα παραγωγικά περιβάλλοντα πρέπει να εφαρμόζουν ένα ξεχωριστό δίκτυο αποθήκευσης (Storage network) για να αυξήσουν την απόδοση και την ασφάλεια.

Μπορούν να αναπτυχθούν περισσότεροι από έναν Block Storage node. (OpenStack, 2017)

3.2.4 Object Storage

Ο προαιρετικός κόμβος αποθήκευσης αντικειμένων (Object Storage node) περιέχει τους δίσκους που χρησιμοποιεί η υπηρεσία αποθήκευσης αντικειμένων (Object Storage service) για την αποθήκευση λογαριασμών, Containers και αντικειμένων (objects).

Για την απλή λειτουργία, η κυκλοφορία υπηρεσιών (service traffic) μεταξύ των Compute nodes και αυτού του κόμβου χρησιμοποιείτε το δίκτυο διαχείρισης (Management network). Τα παραγωγικά περιβάλλοντα πρέπει να εφαρμόζουν ένα ξεχωριστό δίκτυο αποθήκευσης (Storage network) για να αυξήσουν την απόδοση και την ασφάλεια.

Αυτή η υπηρεσία απαιτεί δύο κόμβους και ο κάθε κόμβος απαιτεί τουλάχιστον μία διασύνδεση δικτύου.

Μπορούν να αναπτυχθούν περισσότεροι από δύο Object Storage nodes. (OpenStack, 2017)

3.3 Τα κύρια μέρη του OpenStack

Το OpenStack αποτελείται από πολλά διαφορετικά κινούμενα μέρη. Λόγω της ανοικτής φύσης του, ο καθένας μπορεί να προσθέσει επιπλέον στοιχεία στο OpenStack για να τον βοηθήσει να καλύψει τις ανάγκες του. Όμως, η κοινότητα OpenStack έχει προσδιορίσει σε συνεργασία εννέα βασικά στοιχεία που αποτελούν μέρος του "πυρήνα" του OpenStack, τα οποία διανέμονται ως μέρος οποιουδήποτε συστήματος OpenStack και διατηρούνται επίσημα από την κοινότητα του OpenStack. (Red Hat, Inc, n.d.)

- **Nova**

Είναι ο κύριος υπολογιστής που λειτουργεί πίσω από το OpenStack. Χρησιμοποιείται για την ανάπτυξη και τη διαχείριση μεγάλου αριθμού εικονικών μηχανών και άλλων περιπτώσεων για την εκτέλεση διεργασιών.

- **Swift**

Το Swift είναι ένα σύστημα αποθήκευσης αντικειμένων και αρχείων. Αντί της παραδοσιακής ιδέας της αναφοράς σε αρχεία από τη θέση τους σε μια μονάδα δίσκου, οι προγραμματιστές μπορούν αντ' αυτού να αναφερθούν σε ένα μοναδικό αναγνωριστικό που αναφέρει το αρχείο ή μια πληροφορία και να αφήσει το OpenStack να αποφασίσει πού να αποθηκεύσει αυτές τις πληροφορίες. Αυτό κάνει την κλιμάκωση εύκολη, καθώς οι προγραμματιστές δεν έχουν την ανησυχία για την χωρητικότητα στο σύστημα πίσω από το λογισμικό. Επιτρέπει επίσης στο σύστημα και όχι στον προγραμματιστή να ανησυχεί για τον καλύτερο τρόπο διασφάλισης ότι τα δεδομένα υποστηρίζονται σε περίπτωση σφάλματος μιας μηχανής ή σύνδεσης δικτύου.

- **Cinder**

Το Cinder είναι ένα στοιχείο του block storage (μπλοκ αποθήκευσης), το οποίο είναι πιο κοντά με την παραδοσιακή έννοια του υπολογιστή ως φυσικό μηχάνημα που μπορεί να αποκτήσει πρόσβαση σε συγκεκριμένες θέσεις σε μια μονάδα δίσκου. Αυτός ο πιο παραδοσιακός τρόπος πρόσβασης σε αρχεία μπορεί να είναι σημαντικός σε σενάρια στα οποία η ταχύτητα πρόσβασης δεδομένων είναι η πιο σημαντική παρατήρηση.

- **Neutron**

Το Neutron παρέχει τη δυνατότητα δικτύωσης για το OpenStack. Βοηθά να διασφαλιστεί ότι κάθε ένα από τα κομμάτια ανάπτυξης OpenStack μπορεί να επικοινωνεί μεταξύ τους γρήγορα και αποτελεσματικά.

- **Horizon**

Το dashboard πίσω από το OpenStack ονομάζεται Horizon. Είναι το μόνο γραφικό περιβάλλον του OpenStack, έτσι ώστε οι χρήστες που θέλουν να δοκιμάσουν το OpenStack, μπορεί να είναι το πρώτο μέρος που πραγματικά βλέπουν. Οι προγραμματιστές μπορούν να έχουν πρόσβαση σε όλα τα μέρη του OpenStack ξεχωριστά μέσω interface (API), αλλά ο πίνακας ελέγχου παρέχει στους διαχειριστές συστημάτων να δουν τι συμβαίνει στο Cloud και να το διαχειριστούν όπως είναι απαραίτητο.

- **Keystone**

Το Keystone παρέχει υπηρεσίες ταυτότητας (identity services) για το OpenStack. Είναι ουσιαστικά μια κεντρική λίστα όλων των χρηστών του Cloud OpenStack, καταχωρημένων σε σχέση με τις υπηρεσίες που παρέχονται από το Cloud, τις οποίες έχουν άδεια χρήσης. Παρέχει πολλαπλά μέσα πρόσβασης, που σημαίνει ότι οι προγραμματιστές μπορούν εύκολα να αντιληφθούν και να καταγράψουν τις υπάρχουσες μεθόδους πρόσβασης των χρηστών τους ενάντια στο Keystone.

- **Glance**

Το Glance παρέχει image services στο OpenStack. Σε αυτή την περίπτωση, τα "images" αφορούν images (ή εικονικά αντίγραφα) σκληρών δίσκων. Το Glance επιτρέπει αυτά τα images να χρησιμοποιούνται ως πρότυπα κατά την ανάπτυξη νέων εικονικών μηχανών (virtual machine instances).

- **Ceilometer**

Το Ceilometer παρέχει υπηρεσίες τηλεμετρίας (telemetry services), οι οποίες επιτρέπουν στο Cloud να παρέχει υπηρεσίες χρέωσης σε μεμονωμένους χρήστες του Cloud. Διατηρεί επίσης μια επαληθεύσιμη καταμέτρηση της χρήσης του συστήματος κάθε χρήστη για κάθε ένα από τα διάφορα μέρη ενός Cloud OpenStack.

- **Heat**

Το συστατικό ενορχήστρωσης του OpenStack ονομάζεται Heat , το οποίο επιτρέπει στους προγραμματιστές να αποθηκεύουν τις απαιτήσεις μιας Cloud εφαρμογής σε ένα αρχείο που καθορίζει ποιοι πόροι είναι απαραίτητοι για αυτήν την εφαρμογή. Με αυτόν τον τρόπο, βοηθά στη διαχείριση της υποδομής που απαιτείται για την εκτέλεση μιας Cloud υπηρεσίας. (Red Hat, Inc, n.d.)

3.4 Πώς χρησιμοποιείται το OpenStack σε Cloud περιβάλλον;

Το Cloud αφορά την παροχή υπολογιστικών πόρων στους τελικούς χρήστες σε απομακρυσμένο περιβάλλον, όπου το πραγματικό λογισμικό λειτουργεί ως υπηρεσία σε αξιόπιστους και κλιμακούμενους διακομιστές και όχι σε κάθε υπολογιστή του τελικού χρήστη. Το Cloud computing μπορεί να αναφέρεται σε πολλά διαφορετικά πράγματα, αλλά συνήθως η βιομηχανία μιλά για τη λειτουργία διαφορετικών αντικειμένων ως λογισμικό, πλατφόρμες και υποδομή ως υπηρεσία. Το OpenStack εμπίπτει στην τελευταία κατηγορία και θεωρείται Υποδομή ως Υπηρεσία (IaaS). Η παροχή υποδομής σημαίνει ότι το OpenStack διευκολύνει τους χρήστες να προσθέσουν γρήγορα μια νέα παρουσία (Instance), πάνω στην οποία μπορούν να εκτελεστούν και άλλα στοιχεία του Cloud. Τυπικά, η υποδομή τρέχει τότε μια "πλατφόρμα", πάνω στην οποία ένας προγραμματιστής μπορεί να δημιουργήσει εφαρμογές λογισμικού που παραδίδονται στους τελικούς χρήστες. (Red Hat, Inc, n.d.)

4. Εγκατάσταση και διαμόρφωση OpenStack

Στο κεφάλαιο που ακολουθεί περιγράφεται η δημιουργία μιας ελάχιστης υποδομής Cloud με βάση το OpenStack και την έκδοση Newton με χρήση των παρακάτω τερματικών.

- Controller IP address - 172.16.10.231
- Compute IP address - 172.16.10.232
- Compute IP address - 172.16.10.233
- ObjectStorage1 IP address - 172.16.10.234
- ObjectStorage2 IP address - 172.16.10.235

❖ Controller node

- Physical machine
- Ubuntu 16.04 LTS
- CPU - Intel Core i7 4710q
- 16Gb Ram
- 256Gb SSD
- Realtek LAN driver

❖ Compute node

- Physical machine
- Ubuntu 16.04 LTS
- CPU – Intel Core 2 Duo E6750
- 4Gb Ram
- 250Gb HDD
- Integrated LAN

❖ Block Storage1 node

- Virtual machine – VMWare Workstation 10.0.1
- Ubuntu 16.04 LTS
- CPU – Assigned 3 cores from 1 physical socket
- 4Gb Ram
- 40Gb HDD
- Bridged – Connected directly to the physical network

❖ ObjectStorage1 node

- Virtual machine – VMWare Workstation 10.0.1
- Ubuntu 16.04 LTS
- CPU – Assigned 3 cores from 1 physical socket
- 4Gb Ram
- 40Gb HDD
- Bridged – Connected directly to the physical network

❖ ObjectStorage2 node

- Virtual machine – VMWare Workstation 10.0.1
- Ubuntu 16.04 LTS
- CPU – Assigned 3 cores from 1 physical socket
- 4Gb Ram
- 40Gb HDD
- Bridged – Connected directly to the physical network

4.1 Διαμόρφωση name resolution

Προσθέτουμε τα hostnames των Controller, Network και Compute στην τοποθεσία `/etc/hosts` αντίστοιχα σε όλα τα τερματικά.

❖ Controller node

➤ root@Controller:~# vim /etc/hosts

- # Controller
172.16.10.231 Controller
- # Compute
172.16.10.232 Compute
- # BlockStorage1
172.16.10.233 BlockStorage1
- # ObjectStorage1
172.16.10.234 ObjectStorage1
- # ObjectStorage2
172.16.10.235 ObjectStorage2

```
127.0.0.1 localhost
#127.0.1.1 Controller

#Controller
172.16.10.231 controller

#Compute
172.16.10.232 Compute

#BlockStorage1
172.16.10.233 BlockStorage1

#ObjectStorage1
172.16.10.234 ObjectStorage1
```

❖ Compute node

➤ root@Compute:~# vim /etc/hosts

- # Compute
172.16.10.232 Compute
- # Controller
172.16.10.231 Controller
- # BlockStorage1
172.16.10.233 BlockStorage1
- # ObjectStorage1
172.16.10.234 ObjectStorage1
- # ObjectStorage2
172.16.10.235 ObjectStorage2

```
127.0.0.1 localhost
#127.0.1.1 Compute

#Compute
172.16.10.232 Compute

#Controller
172.16.10.231 Controller

#BlockStorage1
172.16.10.233 BlockStorage1

#ObjectStorage1
172.16.10.234 ObjectStorage1

#ObjectStorage2
172.16.10.235 ObjectStorage2
```

❖ BlockStorage1 node

➤ root@BlockStorage1:~# vim /etc/hosts

- # BlockStorage1
172.16.10.233 BlockStorage1
- # Controller
172.16.10.231 Controller
- # Compute
172.16.10.232 Compute
- # ObjectStorage1
172.16.10.234 ObjectStorage1
- # ObjectStorage2
172.16.10.235 ObjectStorage2

```
127.0.0.1    localhost
#127.0.1.1   BlockStorage1

#BlockStorage1
172.16.10.233 BlockStorage1

#Controller
172.16.10.231 Controller

#Compute
172.16.10.232 Compute

#ObjectStorage1
172.16.10.234 ObjectStorage1

#ObjectStorage2
172.16.10.235 ObjectStorage2
```

❖ ObjectStorage1 node

➤ root@ObjectStorage1:~# vim /etc/hosts

- # ObjectStorage1
172.16.10.234 ObjectStorage1
- # Controller
172.16.10.231 Controller
- # Compute
172.16.10.232 Compute
- # BlockStorage1
172.16.10.233 BlockStorage1
- # ObjectStorage2
172.16.10.235 ObjectStorage2

```
127.0.0.1    localhost
#127.0.1.1   ObjectStorage1

#ObjectStorage1
172.16.10.234 ObjectStorage1

#Controller
172.16.10.231 Controller

#Compute
172.16.10.232 Compute

#BlockStorage1
172.16.10.233 BlockStorage1

#ObjectStorage2
172.16.10.235 ObjectStorage2
```

❖ ObjectStorage2 node

➤ root@ObjectStorage2:~# vim /etc/hosts

- # ObjectStorage2
172.16.10.235 ObjectStorage2
- # Controller
172.16.10.231 Controller
- # Compute
172.16.10.232 Compute
- # BlockStorage1
172.16.10.233 BlockStorage1
- # ObjectStorage1
172.16.10.234 ObjectStorage1

```
127.0.0.1 localhost
#127.0.1.1 ObjectStorage1

#ObjectStorage2
172.16.10.235 ObjectStorage2

#Controller
172.16.10.231 Controller

#Compute
172.16.10.232 Compute

#BlockStorage1
172.16.10.233 BlockStorage1

#ObjectStorage1
172.16.10.234 ObjectStorage1
```

4.2 Networking

Μετά την εγκατάσταση του λειτουργικού συστήματος σε κάθε κόμβο για την αρχιτεκτονική που επιλέξαμε, απενεργοποιούμε το firewall και προχωράμε χειροκίνητα στις ρυθμίσεις που απαιτεί το δίκτυο.

❖ Controller node

Παραμετροποίηση του αρχείου /etc/network/interfaces

- root@Controller:~# vim /etc/network/interfaces
(guide, n.d.)
- IP address: 172.16.10.231
 - Network mask: 255.255.255.0 (or /24)
 - Network: 172.16.10.0
 - Broadcast: 172.16.10.255

- Default gateway: 172.16.10.1
- Dns-nameservers: 8.8.8.8

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto enp5s0f1
iface enp5s0f1 inet static
    address 172.16.10.231
    netmask 255.255.255.0
    network 172.16.10.0
    broadcast 172.16.10.255
    gateway 172.16.10.1
    dns-nameservers 8.8.8.8
```

❖ Compute node

Παραμετροποίηση του αρχείου `/etc/network/interfaces`

➤ `root@Compute:~# vim /etc/network/interfaces`

- IP address: 172.16.10.232
- Network mask: 255.255.255.0 (or /24)
- Network: 172.16.10.0
- Broadcast: 172.16.10.255
- Default gateway: 172.16.10.1
- Dns-nameservers: 8.8.8.8

```
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback

auto enp2s0
iface enp2s0 inet static
    address 172.16.10.232
    netmask 255.255.255.0
    network 172.16.10.0
    broadcast 172.16.10.255
    gateway 172.16.10.1
    dns-nameservers 8.8.8.8
```

❖ BlockStorage1 node

Παραμετροποίηση του αρχείου `/etc/network/interfaces`

➤ `root@BlockStorage1:~# vim /etc/network/interfaces`

- IP address: 172.16.10.233
- Network mask: 255.255.255.0 (or /24)
- Network: 172.16.10.0
- Broadcast: 172.16.10.255

- Default gateway: 172.16.10.1
- Dns-nameservers: 8.8.8.8

```
## This file describes the network interfaces available on your system
## and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet static
    address 172.16.10.233
    netmask 255.255.255.0
    network 172.16.10.0
    broadcast 172.16.10.255
    gateway 172.16.10.1
    dns-nameservers 8.8.8.8
```

❖ ObjectStorage1 node

Παραμετροποίηση του αρχείου `/etc/network/interfaces`

➤ `root@ObjectStorage1:~# vim /etc/network/interfaces`

- IP address: 172.16.10.234
- Network mask: 255.255.255.0 (or /24)
- Network: 172.16.10.0
- Broadcast: 172.16.10.255
- Default gateway: 172.16.10.1
- Dns-nameservers: 8.8.8.8

```
## This file describes the network interfaces available on your system
## and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens33
iface ens33 inet static
    address 172.16.10.234
    netmask 255.255.255.0
    network 172.16.10.0
    broadcast 172.16.10.255
    gateway 172.16.10.1
    dns-nameservers 8.8.8.8
```

🚦 Επανεκκίνηση των τερματικών ώστε να πάρουνε τις ρυθμίσεις.

4.2.1 Επαλήθευσης σύνδεσης των κόμβων

❖ Controller node

➤ root@Controller:~# ping -c 4 www.google.com

```
root@Controller:~# ping -c 4 www.google.com
PING www.google.com (216.58.209.4) 56(84) bytes of data.
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=1 ttl=54 time=22.6 ms
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=2 ttl=54 time=22.6 ms
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=3 ttl=54 time=22.3 ms
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=4 ttl=54 time=22.4 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 22.334/22.523/22.648/0.131 ms
```

➤ root@Controller:~# ping -c 4 Compute

```
root@Controller:~# ping -c 4 Compute
PING Compute (172.16.10.232) 56(84) bytes of data.
64 bytes from Compute (172.16.10.232): icmp_seq=1 ttl=64 time=0.184 ms
64 bytes from Compute (172.16.10.232): icmp_seq=2 ttl=64 time=0.187 ms
64 bytes from Compute (172.16.10.232): icmp_seq=3 ttl=64 time=0.213 ms
64 bytes from Compute (172.16.10.232): icmp_seq=4 ttl=64 time=0.193 ms

--- Compute ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.184/0.194/0.213/0.015 ms
```

➤ root@Controller:~# ping -c 4 BlockStorage1

```
root@Controller:~# ping -c 4 BlockStorage1
PING BlockStorage1 (172.16.10.233) 56(84) bytes of data.
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=1 ttl=64 time=1.39 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=2 ttl=64 time=3.20 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=3 ttl=64 time=1.58 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=4 ttl=64 time=6.64 ms

--- BlockStorage1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 1.395/3.205/6.644/2.106 ms
```

➤ root@Controller:~# ping -c 4 ObjectStorage1

```
root@Controller:~# ping -c 4 ObjectStorage1
PING ObjectStorage1 (172.16.10.234) 56(84) bytes of data.
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=1 ttl=64 time=2.80 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=2 ttl=64 time=1.80 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=3 ttl=64 time=1.37 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=4 ttl=64 time=2.88 ms

--- ObjectStorage1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.370/2.217/2.885/0.648 ms
```

❖ Compute node

➤ root@Compute:~# ping -c 4 www.google.com

```
root@Compute:~# ping -c 4 www.google.com
PING www.google.com (216.58.208.100) 56(84) bytes of data.
64 bytes from sof01s11-in-f100.1e100.net (216.58.208.100): icmp_seq=1 ttl=54 time=22.5 ms
64 bytes from sof01s11-in-f100.1e100.net (216.58.208.100): icmp_seq=2 ttl=54 time=22.5 ms
64 bytes from sof01s11-in-f100.1e100.net (216.58.208.100): icmp_seq=3 ttl=54 time=22.3 ms
64 bytes from sof01s11-in-f100.1e100.net (216.58.208.100): icmp_seq=4 ttl=54 time=22.3 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 22.378/22.481/22.584/0.207 ms
```

➤ root@Compute:~# ping -c 4 Controller

```
root@Compute:~# ping -c 4 Controller
PING Controller (172.16.10.231) 56(84) bytes of data.
64 bytes from Controller (172.16.10.231): icmp_seq=1 ttl=64 time=0.273 ms
64 bytes from Controller (172.16.10.231): icmp_seq=2 ttl=64 time=0.262 ms
64 bytes from Controller (172.16.10.231): icmp_seq=3 ttl=64 time=0.233 ms
64 bytes from Controller (172.16.10.231): icmp_seq=4 ttl=64 time=0.214 ms

--- Controller ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.214/0.245/0.273/0.028 ms
```

➤ root@Compute:~# ping -c 4 BlockStorage1

```
root@Compute:~# ping -c 4 BlockStorage1
PING BlockStorage1 (172.16.10.233) 56(84) bytes of data.
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=1 ttl=64 time=1.47 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=2 ttl=64 time=1.61 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=3 ttl=64 time=3.38 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=4 ttl=64 time=6.22 ms

--- BlockStorage1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.475/3.174/6.221/1.914 ms
```

➤ root@Compute:~# ping -c 4 ObjectStorage1

```
root@Compute:~# ping -c 4 ObjectStorage1
PING ObjectStorage1 (172.16.10.234) 56(84) bytes of data.
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=1 ttl=64 time=1.54 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=2 ttl=64 time=1.55 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=3 ttl=64 time=1.75 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=4 ttl=64 time=2.79 ms

--- ObjectStorage1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.542/1.911/2.798/0.521 ms
```

❖ BlockStorage1 node

➤ root@BlockStorage1:~# ping -c 4 www.google.com

```
root@BlockStorage1:~# ping -c 4 www.google.com
PING www.google.com (172.217.17.164) 56(84) bytes of data.
64 bytes from sof02s21-in-f164.1e100.net (172.217.17.164): icmp_seq=1 ttl=55 time=18.1 ms
64 bytes from sof02s21-in-f164.1e100.net (172.217.17.164): icmp_seq=2 ttl=55 time=22.7 ms
64 bytes from sof02s21-in-f164.1e100.net (172.217.17.164): icmp_seq=3 ttl=55 time=17.2 ms
64 bytes from sof02s21-in-f164.1e100.net (172.217.17.164): icmp_seq=4 ttl=55 time=17.4 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 17.229/18.886/22.749/2.261 ms
```

- root@BlockStorage1:~# ping -c 4 Controller

```
root@BlockStorage1:~# ping -c 4 Controller
PING Controller (172.16.10.231) 56(84) bytes of data.
64 bytes from Controller (172.16.10.231): icmp_seq=1 ttl=64 time=3.66 ms
64 bytes from Controller (172.16.10.231): icmp_seq=2 ttl=64 time=1.43 ms
64 bytes from Controller (172.16.10.231): icmp_seq=3 ttl=64 time=2.64 ms
64 bytes from Controller (172.16.10.231): icmp_seq=4 ttl=64 time=3.47 ms

--- Controller ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 1.434/2.806/3.668/0.881 ms
```

- root@BlockStorage1:~# ping -c 4 Compute

```
root@BlockStorage1:~# ping -c 4 Compute
PING Compute (172.16.10.232) 56(84) bytes of data.
64 bytes from Compute (172.16.10.232): icmp_seq=1 ttl=64 time=3.46 ms
64 bytes from Compute (172.16.10.232): icmp_seq=2 ttl=64 time=1.90 ms
64 bytes from Compute (172.16.10.232): icmp_seq=3 ttl=64 time=2.91 ms
64 bytes from Compute (172.16.10.232): icmp_seq=4 ttl=64 time=1.34 ms

--- Compute ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.340/2.408/3.469/0.833 ms
```

- root@BlockStorage1:~# ping -c 4 ObjectStorage1

```
root@BlockStorage1:~# ping -c 4 ObjectStorage1
PING ObjectStorage1 (172.16.10.234) 56(84) bytes of data.
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=1 ttl=64 time=0.700 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=2 ttl=64 time=0.450 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=3 ttl=64 time=0.449 ms
64 bytes from ObjectStorage1 (172.16.10.234): icmp_seq=4 ttl=64 time=0.694 ms

--- ObjectStorage1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.449/0.573/0.700/0.124 ms
```

❖ ObjectStorage1 node

- root@ObjectStorage1:~# ping -c 4 www.google.com

```
root@ObjectStorage1:~# ping -c 4 www.google.com
PING www.google.com (216.58.209.4) 56(84) bytes of data.
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=1 ttl=54 time=25.4 ms
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=2 ttl=54 time=27.3 ms
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=3 ttl=54 time=23.8 ms
64 bytes from sof01s12-in-f4.1e100.net (216.58.209.4): icmp_seq=4 ttl=54 time=23.8 ms

--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 23.840/25.133/27.361/1.445 ms
```

- root@ObjectStorage1:~# ping -c 4 Controller

```
root@ObjectStorage1:~# ping -c 4 Controller
PING Controller (172.16.10.231) 56(84) bytes of data.
64 bytes from Controller (172.16.10.231): icmp_seq=1 ttl=64 time=1.55 ms
64 bytes from Controller (172.16.10.231): icmp_seq=2 ttl=64 time=8.96 ms
64 bytes from Controller (172.16.10.231): icmp_seq=3 ttl=64 time=8.60 ms
64 bytes from Controller (172.16.10.231): icmp_seq=4 ttl=64 time=2.99 ms

--- Controller ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.552/5.528/8.963/3.297 ms
```

➤ root@ObjectStorage1:~# ping -c 4 Compute

```
root@ObjectStorage1:~# ping -c 4 Compute
PING Compute (172.16.10.232) 56(84) bytes of data.
64 bytes from Compute (172.16.10.232): icmp_seq=1 ttl=64 time=2.97 ms
64 bytes from Compute (172.16.10.232): icmp_seq=2 ttl=64 time=3.09 ms
64 bytes from Compute (172.16.10.232): icmp_seq=3 ttl=64 time=1.32 ms
64 bytes from Compute (172.16.10.232): icmp_seq=4 ttl=64 time=3.04 ms

--- Compute ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.325/2.608/3.094/0.744 ms
```

➤ root@ObjectStorage1:~# ping -c 4 BlockStorage1

```
root@ObjectStorage1:~# ping -c 4 BlockStorage1
PING BlockStorage1 (172.16.10.233) 56(84) bytes of data.
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=1 ttl=64 time=0.652 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=2 ttl=64 time=0.525 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=3 ttl=64 time=1.14 ms
64 bytes from BlockStorage1 (172.16.10.233): icmp_seq=4 ttl=64 time=0.687 ms

--- BlockStorage1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.525/0.752/1.145/0.235 ms
```

4.3 Network Time Protocol (NTP)

❖ Controller node

➤ root@Controller:~# apt install chrony

Παραμετροποίηση του αρχείου **chrony.conf** στην τοποθεσία **/etc/chrony/chrony.conf** προσθέτοντας τον NTP Server της επιλογής μας.

➤ root@Controller:~# vim /etc/chrony/chrony.conf

- server 0.gr.pool.ntp.org
- server 1.europe.pool.ntp.org
- server 0.europe.pool.ntp.org

```
#pool 2.debian.pool.ntp.org offline iburst
#my ntp servers
server 0.gr.pool.ntp.org
server 1.europe.pool.ntp.org
server 0.europe.pool.ntp.org

# Look here for the admin password needed for chronyc. The initial
# password is generated by a random process at install time. You may
# change it if you wish.

keyfile /etc/chrony/chrony.keys
```

Για να επιτραπεί σε άλλους κόμβους του δικτύου να συνδεθούν με τον chrony daemon στον Controller node, προσθέτουμε το παρακάτω στο αρχείο **/etc/chrony/chrony.conf file**

➤ root@Controller:~# vim /etc/chrony/chrony.conf

- allow 172.16.10.0/24


```
#allow foo.example.net
#allow 10/8
#allow 0/0 (allow access by any IPv4 node)
#allow ::/0 (allow access by any IPv6 node)
allow 172.16.10.0/24
```

➤ root@Controller:~# service chrony restart

```
root@Controller:~# service chrony restart
root@Controller:~#
root@Controller:~#
```

❖ Compute node

➤ root@Compute:~# apt install chrony

Παραμετροποίηση του αρχείου /etc/chrony/chrony.conf ώστε να αναφέρεται στον Controller node.

➤ root@Compute:~# vim /etc/chrony/chrony.conf

▪ server Controller iburst

```
#pool 2.debian.pool.ntp.org offline iburst
server Controller iburst

# Look here for the admin password needed for chronyc. The initial
# password is generated by a random process at install time. You may
# change it if you wish.
```

❖ BlockStorage1 node

➤ root@BlockStorage1:~# apt install chrony

Παραμετροποίηση του αρχείου /etc/chrony/chrony.conf ώστε να αναφέρεται στον Controller node.

➤ root@BlockStorage1:~# vim /etc/chrony/chrony.conf

▪ server Controller iburst

```
#pool 2.debian.pool.ntp.org offline iburst
server Controller iburst

# Look here for the admin password needed for chronyc. The initial
# password is generated by a random process at install time. You may
# change it if you wish.
```

❖ ObjectStorage1 node

➤ root@BlockStorage1:~# apt install chrony

Παραμετροποίηση του αρχείου `/etc/chrony/chrony.conf` ώστε να αναφέρεται στον Controller node.

➤ `root@ObjectStorage1:~# vim /etc/chrony/chrony.conf`

- `server Controller iburst`

```
#pool 2.debian.pool.ntp.org offline iburst
server Controller iburst

Look here for the admin password needed for chronyc. The initial
# password is generated by a random process at install time. You may
# change it if you wish.
```

🚦 Επανεκκίνηση NTP service σε όλους τους κόμβους

4.3.1 Επαλήθευση λειτουργίας του NTP

❖ Controller node

➤ `root@Controller:~# chronyc sources`

```
root@Controller:~# chronyc sources
210 Number of sources = 3
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^? skevos.c.arx.gr          3      6      3      1    -790us[ -790us] +/-  67ms
^? arethusa.tweakers.net    2      6      3      1    +404us[ +404us] +/-  74ms
^? ntp.zeptobars.com        2      6      3      2    -15ms[ -15ms] +/- 257ms
root@Controller:~#
```

✓ Γράφοντας `date` μας εμφανίζει την ώρα και την ημερομηνία που έχει το δίκτυο μας μετά τις αλλαγές

➤ `root@Controller:~# date`

```
root@Controller:~# date
Fri Jul 28 01:12:11 EEST 2017
root@Controller:~#
```

❖ Compute node

➤ `root@Controller:~# chronyc sources`

```
root@Compute:~# chronyc sources
210 Number of sources = 1
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* Controller                4      6      3     53   +754us[+4857us] +/-  67ms
root@Compute:~#
```

✓ Γράφοντας `date` μας εμφανίζει την ώρα και την ημερομηνία που έχει το δίκτυο μας μετά τις αλλαγές

➤ `root@Compute:~# date`


```
root@Compute:~# date
Παρ 28 Ιούλ 2017 01:14:06 πμ EEST
root@Compute:~#
```

❖ BlockStorage1 node

➤ root@BlockStorage1:~# chronyc sources

```
root@BlockStorage1:~# chronyc sources
210 Number of sources = 1
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* Controller                3   6   377   44  +1397us[+1633us] +/-  44ms
root@BlockStorage1:~#
```

✓ Γράφοντας **date** μας εμφανίζει την ώρα και την ημερομηνία που έχει το δίκτυο μας μετά τις αλλαγές

➤ root@Network:~# date

```
root@BlockStorage1:~# date
Thu Jul 27 23:09:08 EEST 2017
root@BlockStorage1:~#
root@BlockStorage1:~#
```

❖ ObjectStorage1 node

➤ root@ObjectStorage1:~# chronyc sources

```
root@ObjectStorage1:~# chronyc sources
210 Number of sources = 1
MS Name/IP address          Stratum Poll Reach LastRx Last sample
=====
^* Controller                3   7   377   11  +2229us[+2305us] +/-  46ms
root@ObjectStorage1:~#
```

✓ Γράφοντας **date** μας εμφανίζει την ώρα και την ημερομηνία που έχει το δίκτυο μας μετά τις αλλαγές

➤ root@ObjectStorage1:~# date

```
root@ObjectStorage1:~# date
Thu Jul 27 23:09:26 EEST 2017
root@ObjectStorage1:~#
root@ObjectStorage1:~#
```

4.4 OpenStack packages

✓ Η εγκατάσταση των πακέτων OpenStack πρέπει να γίνει σε όλους τους κόμβους: **Controller**, **Compute**, **BlockStorage1**, **ObjStorage1** και **ObjectStorage2** απενεργοποιώντας τις αυτόματες ενημερώσεις για να μην επηρεάσουν το σύστημα μας.

4.4.1 Ενεργοποίηση του OpenStack repository

Τρέχουμε τις παρακάτω εντολές για όλους τους κόμβους του δικτύου

- root@Controller:~# apt install software-properties-common
- root@Controller:~# add-apt-repository cloud-archive:newton

- root@Compute:~# apt install software-properties-common
- root@Compute:~# add-apt-repository cloud-archive:newton

- root@BlockStorage1:~# apt install software-properties-common
- root@BlockStorage1:~# add-apt-repository cloud-archive:newton

- root@ObjectStorage1:~# apt install software-properties-common
- root@ObjectStorage1:~# add-apt-repository cloud-archive:newton

- root@ObjectStorage2:~# apt install software-properties-common
- root@ObjectStorage2:~# add-apt-repository cloud-archive:newton

4.4.2 Ολοκλήρωση εγκατάστασης

Upgrade packages

- root@Controller:~# apt update && apt dist-upgrade
 - root@Compute:~# apt update && apt dist-upgrade
 - root@BlockStorage1:~# apt update && apt dist-upgrade
 - root@ObjectStorage1:~# apt update && apt dist-upgrade
 - root@ObjectStorage2:~# apt update && apt dist-upgrade
- ✓ Εάν η διαδικασία αναβάθμισης περιλαμβάνει νέο Kernel, επανεκκινήστε το τερματικό για την ενεργοποίησή του.

Εγκατάσταση OpenStack Client

- root@Controller:~# apt install python-openstackclient
- root@Compute:~# apt install python-openstackclient
- root@BlockStorage1:~# apt install python-openstackclient
- root@ObjectStorage1:~# apt install python-openstackclient
- root@ObjectStorage2:~# apt install python-openstackclient

4.5 Database

Οι περισσότερες υπηρεσίες OpenStack χρησιμοποιούν μια βάση δεδομένων SQL για την αποθήκευση πληροφοριών η οποία συνήθως τρέχει στον Controller node.

Σε αυτόν τον οδηγό θα χρησιμοποιηθούν αναλόγως, MariaDB ή MySQL. Επίσης οι υπηρεσίες OpenStack υποστηρίζουν επίσης και άλλες βάσεις δεδομένων SQL, όπως την PostgreSQL. (guide, n.d.)

4.5.1 Εγκατάσταση πακέτων στον Controller node

➤ root@Controller:~# apt install mariadb-server python-pymysql

Δημιουργία και παραμετροποίηση του αρχείου /etc/mysql/mariadb.conf.d/99-openstack.cnf

➤ root@Controller:~# vim /etc/mysql/mariadb.conf.d/99-openstack.cnf

```
[mysqld]
bind-address = 172.16.10.231
default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

```
[mysqld]
bind-address = 172.16.10.231

default-storage-engine = innodb
innodb_file_per_table
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

4.5.2 Ολοκλήρωση εγκατάστασης

🔧 Επανεκκίνηση mysql service

➤ root@Controller:~# service mysql restart

🔧 Εκτέλεση του script mysql_secure_installation για την ασφάλεια της βάσης δεδομένων

➤ root@Controller:~# mysql_secure_installation

- Enter current password for root (enter for none): **123456**
- Change the root password? [Y/n]: **n**
- Remove anonymous users? [Y/n]: **y**
- Disallow root login remotely? [Y/n]: **y**
- Remove test database and access to it? [Y/n]: **y**
- Reload privilege tables now? [Y/n]: **y**

```
root@Controller:~# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
root@Controller:~#
```

4.6 Διακομιστής μηνυμάτων - Messaging server

Το OpenStack χρησιμοποιεί έναν message broker για τον συντονισμό των λειτουργιών και της κατάστασης των πληροφοριών μεταξύ των υπηρεσιών.

Η υπηρεσία message broker τυπικά τρέχει στον Controller node. Το OpenStack υποστηρίζει πολλούς message brokers συμπεριλαμβανομένων των **RabbitMQ**, **Qpid** και **ZeroMQ**. Ωστόσο, οι περισσότερες διανομές που περιέχουν το OpenStack υποστηρίζουν έναν συγκεκριμένο message broker. Αυτός ο οδηγός καλύπτει τον message broker RabbitMQ, ο οποίος υποστηρίζεται από κάθε διανομή. (guide, n.d.)

4.6.1 Εγκατάσταση και διαμορφώση

- root@Controller:~# apt install rabbitmq-server
- ✓ Προσθέτουμε τον χρήστη openstack
- root@Controller:~# rabbitmqctl add_user openstack RABBIT_PASS
- ✓ add_user: openstack
- ✓ RABBIT_PASS: 123456

```
root@Controller:~# rabbitmqctl add_user openstack 123456
Creating user "openstack" ...
root@Controller:~#
```

- ✓ Δικαιώματα για τον χρήστη, read - write
- root@Controller:~# rabbitmqctl set_permissions openstack ".*" ".*" ".*"

Setting permissions for user "openstack" in vhost "/" ...

```
root@Controller:~# rabbitmqctl set_permissions openstack ".*" ".*" ".*"
Setting permissions for user "openstack" in vhost "/" ...
root@Controller:~#
```

- ✓ Ανοίγουμε την προεπιλεγμένη πόρτα 5672 στο δίκτυο
- root@Controller:~# vim /etc/rabbitmq/rabbitmq-env.conf

```
# Defaults to rabbit. This can be useful if you want to run more than one node
# per machine - RABBITMQ_NODENAME should be unique per erlang-node-and-machine
# combination. See the clustering on a single machine guide for details:
# http://www.rabbitmq.com/clustering.html#single-machine
#NODENAME=rabbit

# By default RabbitMQ will bind to all interfaces, on IPv4 and IPv6 if
# available. Set this if you only want to bind to one network interface or#
# address family.
#NODE_IP_ADDRESS=127.0.0.1

# Defaults to 5672.
#NODE_PORT=5672
```

Επανεκκίνηση service rabbitmq-server

- root@Controller:~# service rabbitmq-server restart

4.7 Memcached

Ο μηχανισμός Identity service authentication για υπηρεσίες χρησιμοποιεί Memcached to cache tokens και συνήθως τρέχει στον Controller node.

Για τις υλοποιήσεις παραγωγής, ενεργοποιούμε ένα συνδυασμό τείχους προστασίας, ελέγχου ταυτότητας και κρυπτογράφησης για να το ασφαλίσουμε. (guide, n.d.)

4.7.1 Εγκατάσταση και διαμόρφωση

Εγκατάσταση πακέτων

➤ root@Controller:~# apt install memcached python-memcache

🔧 Επανεκκινούμε το service **memcached**

➤ root@Controller:~# service memcached restart

Παραμετροποίηση του αρχείου **memcached.conf** στην τοποθεσία **/etc/memcached.conf** και διαμόρφωση της υπηρεσίας να χρησιμοποιεί την **management IP address**.

➤ root@Controller:~# vim /etc/memcached.conf

-l 127.0.0.1 → -l 172.16.10.231

```
# Run memcached as a daemon. This command is implied, and is not needed for the
# daemon to run. See the README.Debian that comes with this package for more
# information.
-d

# Log memcached's output to /var/log/memcached
logfile /var/log/memcached.log

# Be verbose
# -v

# Be even more verbose (print client commands as well)
# -vv

# Start with a cap of 64 megs of memory. It's reasonable, and the daemon default
# Note that the daemon will grow to this size, but does not start out holding this much
# memory
-m 64

# Default connection port is 11211
-p 11211

# Run the daemon as root. The start-memcached will default to running as root if no
# -u command is present in this config file
-u memcache

# Specify which IP address to listen on. The default is to listen on all IP addresses
# This parameter is one of the only security measures that memcached has, so make sure
# it's listening on a firewalled interface.
-l 172.16.10.231
```

4.7.2 Ολοκλήρωση εγκατάστασης

🔧 Επανεκκίνησή το service **memcached**

➤ root@Controller:~# service memcached restart

4.8 Υπηρεσία ταυτότητας - Identity service

Αυτή η ενότητα περιγράφει τον τρόπο εγκατάστασης και ρύθμισης του OpenStack Identity service, με κωδικό όνομα, στον Controller node.

Για λόγους κλιμάκωσης, αυτή η διαμόρφωση αναπτύσσει Fernet και τον διακομιστή HTTP Apache για να χειρίζονται τα αιτήματα.

Η υπηρεσία OpenStack Identity εκτελεί τις ακόλουθες λειτουργίες:

- Παρακολούθηση χρηστών και των αδειών τους.
- Παροχή καταλόγου διαθέσιμων υπηρεσιών με τα API endpoints.

Κατά την εγκατάσταση της υπηρεσίας OpenStack Identity, πρέπει να καταχωρηθεί κάθε υπηρεσία στην εγκατάσταση OpenStack.

Το Identity service μπορεί στη συνέχεια να παρακολουθήσει ποιες υπηρεσίες OpenStack έχουν εγκατασταθεί και βρίσκονται στο δίκτυο. (guide, n.d.)

4.8.1 Έννοιες OpenStack Identity Service

User

- ✓ Η ψηφιακή αναπαράσταση ενός ατόμου, συστήματος ή υπηρεσίας που χρησιμοποιεί υπηρεσίες Cloud OpenStack.
- ✓ Το Identity service επικυρώνει τις αιτήσεις που γίνονται από το χρήστη που κάνει την κλήση.
- ✓ Οι χρήστες έχουν μια σύνδεση και μπορεί να τους δοθούν tokens για πρόσβαση σε πόρους.
- ✓ Οι χρήστες μπορούν να εκχωρηθούν απευθείας σε ένα συγκεκριμένο tenant και να συμπεριφέρονται σαν να περιέχονται στο συγκεκριμένο tenant.

Credentials

Δεδομένα που επιβεβαιώνουν την ταυτότητα του χρήστη.

Για παράδειγμα: όνομα χρήστη και κωδικό πρόσβασης, όνομα χρήστη και API key ή ένα διακριτικό ελέγχου ταυτότητας που παρέχεται από το Identity Service.

Authentication

Η διαδικασία επιβεβαίωσης της ταυτότητας ενός χρήστη.

Το OpenStack Identity επιβεβαιώνει μια εισερχόμενη αίτηση επικυρώνοντας ένα σύνολο διαπιστευτηρίων που παρέχονται από τον χρήστη.

Αυτά τα διαπιστευτήρια είναι αρχικά ένα όνομα χρήστη και κωδικός πρόσβασης ή ένα όνομα χρήστη και API key.

Όταν τα διαπιστευτήρια χρήστη έχουν επικυρωθεί, το OpenStack Identity εκδίδει ένα authentication token το οποίο ο χρήστης παρέχει σε επόμενα αιτήματα.

Token

Μια αλφαριθμητική σειρά κειμένου που χρησιμοποιείται για την πρόσβαση σε API και πόρους του OpenStack. Ένα token μπορεί να ανακληθεί ανά πάσα στιγμή και ισχύει για μια πεπερασμένη διάρκεια.

Παρόλο που το OpenStack Identity υποστηρίζει τον έλεγχο ταυτότητας που βασίζεται σε token σε αυτήν την έκδοση, η πρόθεση είναι να υποστηριχθούν πρόσθετα πρωτόκολλα στο μέλλον. Ο κύριος σκοπός του είναι να είναι μια υπηρεσία ενσωμάτωσης και δεν φιλοδοξεί να είναι μια ολοκληρωμένη λύση αποθήκευσης και διαχείρισης ταυτότητας.

Tenant

- ✓ Ένα «δοχείο» που χρησιμοποιείται για την ομαδοποίηση ή την απομόνωση πόρων.
- ✓ Τα Tenants ομαδοποιούν επίσης ή απομονώνουν identity objects.
- ✓ Ανάλογα με τον φορέα παροχής υπηρεσιών, ένα tenant μπορεί να αντιστοιχίσει έναν πελάτη, έναν λογαριασμό, μια οργάνωση ή ένα έργο.

Service

Μια υπηρεσία OpenStack, όπως Compute (nova), Object Storage (swift) ή Image Service (glance). Παρέχει ένα ή περισσότερα endpoints στα οποία οι χρήστες μπορούν να έχουν πρόσβαση σε πόρους και να εκτελούν εργασίες.

Endpoint

Μια διεύθυνση προσπελάσιμη από το δίκτυο, όπου υπάρχει πρόσβαση σε μια υπηρεσία, συνήθως μια διεύθυνση URL.

Εάν χρησιμοποιείτε μια επέκταση για πρότυπα, μπορεί να δημιουργηθεί ένα πρότυπο endpoint, το οποίο αντιπροσωπεύει τα πρότυπα όλων των αναλώσιμων υπηρεσιών που είναι διαθέσιμες σε όλες τις περιοχές.

Role

Μια προσωπικότητα με καθορισμένο σύνολο δικαιωμάτων και προνομίων για να εκτελέσει ένα συγκεκριμένο σύνολο λειτουργιών.

Στο Identity service, ένα token που εκδίδεται σε ένα χρήστη περιλαμβάνει τη λίστα ρόλων. Οι υπηρεσίες που καλούνται από αυτόν καθορίζουν πώς ερμηνεύουν το σύνολο των ρόλων που έχει ένας χρήστης και σε ποιες λειτουργίες ή πόρους κάθε ρόλος παρέχει πρόσβαση.

Keystone Client

Μια διεπαφή γραμμής εντολών για το OpenStack Identity API.

Για παράδειγμα, οι χρήστες μπορούν να εκτελέσουν εντολές για την καταχώρηση των υπηρεσιών στις εγκαταστάσεις τους OpenStack. (guide, n.d.)

4.8.2 Η υπηρεσία "Identity service" περιέχει τα παρακάτω στοιχεία

Server

Ο κεντρικός διακομιστής παρέχει υπηρεσίες ελέγχου ταυτότητας και εξουσιοδότησης χρησιμοποιώντας μια διεπαφή RESTful.

Drivers

Οι Drivers είναι ενσωματωμένοι στον κεντρικό διακομιστή.

Χρησιμοποιούνται για την πρόσβαση σε πληροφορίες ταυτότητας σε εξωτερικά repositories του OpenStack και ενδέχεται να υπάρχουν ήδη στην υποδομή όπου αναπτύσσεται το OpenStack (για παράδειγμα, βάσεις δεδομένων SQL ή διακομιστές LDAP).

Modules

Τα Modules Middleware εκτελούνται στο χώρο διεύθυνσης του OpenStack που χρησιμοποιεί την Identity service. Αυτές οι ενότητες παρεμποδίζουν τις αιτήσεις παροχής υπηρεσιών, εξαγάγουν διαπιστευτήρια για τον χρήστη και τα αποστέλλουν στον κεντρικό διακομιστή για εξουσιοδότηση.

Η ενσωμάτωση μεταξύ των ενοτήτων ενδιάμεσου λογισμικού και των Modules του OpenStack χρησιμοποιεί το Python Web Server Gateway Interface. (guide, n.d.)

4.8.3 Εγκατάσταση και διαμόρφωση

- ✓ Πριν ρυθμίσουμε την υπηρεσία ταυτότητας OpenStack - OpenStack Identity service, πρέπει να δημιουργήσουμε μια βάση δεδομένων και ένα token διαχείρισης.

Παρακάτω περιγράφεται ο τρόπος εγκατάστασης και ρύθμισης παραμέτρων της υπηρεσίας OpenStack Identity στον Controller node.

Δημιουργία βάσης δεδομένων - **keystone**

```
➤ root@Controller:~# mysql -u root -p
```

```
MariaDB [(none)]> CREATE DATABASE keystone;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' \
IDENTIFIED BY 'KEYSTONE_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' \
IDENTIFIED BY 'KEYSTONE_DBPASS';
```

```
MariaDB [(none)]> exit
```

KEYSTONE_DBPASS: **123456**

```

root@Controller:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.0.29-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04
Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]> CREATE DATABASE keystone;
Query OK, 1 row affected (0.00 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)
MariaDB [(none)]> exit
Bye
root@Controller:~#
  
```

🔧 Επανεκκίνηση το service mysql

➤ root@Controller:~# service mysql restart

Εγκαταστήσουμε των πακέτων

➤ root@Controller:~# apt install keystone

Παραμετροποίηση του /etc/keystone/keystone.conf file

➤ root@Controller:~# vim /etc/keystone/keystone.conf

[database]

...

connection = mysql+pymysql://keystone:KEYSTONE_DBPASS@controller/keystone

```

[database]
#
# From oslo.db
#
# DEPRECATED: The file name to use with SQLite. (string value)
# Deprecated group/name - [DEFAULT]/sqlite_db
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Should use config option connection or slave_connection to connect
# the database.
#sqlite_db = oslo.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecated group/name - [DEFAULT]/sqlite_synchronous
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://keystone:123456@controller/keystone
  
```

KEYSTONE_DBPASS: 123456

[token]

...

Provider = fernet

```
[token]
#
# From keystone
#
# This is a list of external authentication mechanisms which should add token
# binding metadata to tokens, such as 'kerberos' or 'x509'. Binding metadata is
# enforced according to the '[token] enforce_token_bind' option. (list value)
#bind =

# This controls the token binding enforcement policy on tokens presented to
# keystone with token binding metadata (as specified by the '[token] bind'
# option). 'disabled' completely bypasses token binding validation.
# 'permissive' and 'strict' do not require tokens to have binding metadata (but
# will validate it if present), whereas 'required' will always demand tokens to
# have binding metadata. 'permissive' will allow unsupported binding metadata
# to pass through without validation (usually to be validated at another time
# by another component), whereas 'strict' and 'required' will demand that the
# included binding metadata be supported by keystone. (string value)
# Allowed values: disabled, permissive, strict, required
#enforce_token_bind = permissive

# The amount of time that a token should remain valid (in seconds). Drastically
# reducing this value may break "long-running" operations that involve multiple
# services to coordinate together, and will force users to authenticate with
# keystone more frequently. Drastically increasing this value will increase
# load on the '[token] driver', as more tokens will be simultaneously valid.
# Keystone tokens are also bearer tokens, so a shorter duration will also
# reduce the potential security impact of a compromised token. (integer value)
# Minimum value: 0
# Maximum value: 9223372036854775807
#expiration = 3600

# Entry point for the token provider in the 'keystone.token.provider'
# namespace. The token provider controls the token construction, validation,
# and revocation operations. Keystone includes 'fernet' and 'uuid' token
# providers. 'uuid' tokens must be persisted (using the backend specified in
# the '[token] driver' option), but do not require any extra configuration or
# setup. 'fernet' tokens do not need to be persisted at all, but require that
# you run 'keystone-manage fernet_setup' (also see the 'keystone-manage
# fernet_rotate' command). (string value)
provider = fernet
```

Populate the Identity service database

➤ root@Controller:~# su -s /bin/sh -c "keystone-manage db_sync" keystone

Αρχικοποίηση Fernet key repositories

➤ root@Controller:~# keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone

➤ root@Controller:~# keystone-manage credential_setup --keystone-user keystone --keystone-group keystone

```
root@Controller:~# keystone-manage fernet_setup --keystone-user keystone --keystone-group keystone
2017-07-08 23:11:06.534 4050 INFO keystone.common.fernet_utils [-] Key repository is already initialized; aborting.
root@Controller:~# keystone-manage credential_setup --keystone-user keystone --keystone-group keystone
2017-07-08 23:11:26.775 4061 INFO keystone.common.fernet_utils [-] Key repository is already initialized; aborting.
```

Bootstrap the Identity service

➤ root@Controller:~# keystone-manage bootstrap --bootstrap-password ADMIN_PASS \
 --bootstrap-admin-url http://controller:35357/v3/\
 --bootstrap-internal-url http://controller:35357/v3/\
 --bootstrap-public-url http://controller:5000/v3/\
 --bootstrap-region-id RegionOne

```

root@Controller:~# keystone-manage bootstrap --bootstrap-password 123456 \
--bootstrap-admin-url http://controller:35357/v3/ \
--bootstrap-internal-url http://controller:35357/v3/ \
--bootstrap-public-url http://controller:5000/v3/ \
--bootstrap-region-id RegionOne
2017-07-08 23:16:34.476 4254 WARNING py.warnings [-] /usr/lib/python2.7/dist-packages/pycadf/Identifier.py:66: UserWarning: Invalid uid. To ensure interoperability, identifiers should be a valid uid.
Warnings.warn('Invalid uid. To ensure interoperability, identifiers '
2017-07-08 23:16:34.662 4254 INFO keystone.cnd.cll [...] Created domain default
2017-07-08 23:16:34.818 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created project admin
2017-07-08 23:16:34.937 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created user admin
2017-07-08 23:16:35.026 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created role admin
2017-07-08 23:16:35.126 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Granted admin on admin to user admin.
2017-07-08 23:16:35.298 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created region RegionOne
2017-07-08 23:16:35.345 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created admin endpoint http://controller:35357/v3/
2017-07-08 23:16:35.583 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created internal endpoint http://controller:35357/v3/
2017-07-08 23:16:35.592 4254 INFO keystone.cnd.cll [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Created public endpoint http://controller:5000/v3/
2017-07-08 23:16:35.564 4254 INFO keystone.assignment.core [req-e762b998-a7cf-4f62-9d3c-719ca7fccc25 - - - -] Creating the default role 9fe2ff9ee4384b1894e98878d3e92bab because it does not exist.

```

ADMIN_PASS: 123456

Διαμόρφωση του Apache HTTP server

Παραμετροποίηση του /etc/apache2/apache2.conf και διαμόρφωση του ServerName να αναφέρεται στον Controller Node

- root@Controller:~# vim /etc/apache2/apache2.conf

ServerName controller

```

AllowOverride None
Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
ServerName Controller

```

4.8.4 Ολοκλήρωση εγκατάστασης

Επανεκκίνηση του Apache service και κατάργηση της προεπιλεγμένη βάση δεδομένων SQLite:

- root@Controller:~# service apache2 restart
- root@Controller:~# rm -f /var/lib/keystone/keystone.db

Διαμόρφωση environment variable για τον admin user

- root@Controller:~# vim keystone_admin.sh

```

$ export OS_USERNAME=admin
$ export OS_PASSWORD=ADMIN_PASS
$ export OS_PROJECT_NAME=admin
$ export OS_USER_DOMAIN_NAME=Default
$ export OS_PROJECT_DOMAIN_NAME=Default
$ export OS_AUTH_URL=http://controller:35357/v3
$ export OS_IDENTITY_API_VERSION=3

```

```

$ export OS_USERNAME=admin
$ export OS_PASSWORD=123456
$ export OS_PROJECT_NAME=admin
$ export OS_USER_DOMAIN_NAME=Default
$ export OS_PROJECT_DOMAIN_NAME=Default
$ export OS_AUTH_URL=http://controller:35357/v3
$ export OS_IDENTITY_API_VERSION=3

```

Αντικατάσταση του **ADMIN_PASS** με τον κωδικό πρόσβασης που χρησιμοποιήσαμε στην εντολή bootstrap από την ενότητα "Εγκατάσταση και ρύθμιση παραμέτρων".

ADMIN_PASS: 123456

Δημιουργία domain, projects, users, and roles

Η υπηρεσία ταυτότητας - Identity service παρέχει υπηρεσίες ελέγχου ταυτότητας για κάθε υπηρεσία OpenStack. Η υπηρεσία ελέγχου ταυτότητας χρησιμοποιεί έναν συνδυασμό domains, projects, users, και ρόλων.

Παρακάτω θα δούμε ότι χρησιμοποιεί ένα Service Project που περιέχει έναν μοναδικό χρήστη για κάθε υπηρεσία που προστίθεται στο περιβάλλον. (guide, n.d.)

Δημιουργία Service Project

- administrator@Controller:~\$ openstack project create --domain default \
--description "Service Project" service

```
administrator@Controller:~$ openstack project create --domain default \  
> --description "Service Project" service  
+-----+  
| Field | Value |  
+-----+  
| description | Service Project |  
| domain_id | default |  
| enabled | True |  
| id | acd9b926e682407c9dfe4a833eb3113b |  
| is_domain | False |  
| name | service |  
| parent_id | default |  
+-----+
```

Δημιουργία demo project

Οι κανονικές (μη διαχειριστικές) εργασίες πρέπει να χρησιμοποιούν ένα μη προνομιούχο έργο και χρήστη. Για παράδειγμα, αυτός ο οδηγός δημιουργεί το demo project και το user.

- administrator@Controller:~\$ openstack project create --domain default \
--description "Demo Project" demo

```
administrator@Controller:~$ openstack project create --domain default \  
> --description "Demo Project" demo  
+-----+  
| Field | Value |  
+-----+  
| description | Demo Project |  
| domain_id | default |  
| enabled | True |  
| id | e2c082fd3fd04b41aa84f90a94208ce8 |  
| is_domain | False |  
| name | demo |  
| parent_id | default |  
+-----+
```

Δημιουργία χρήστη Demo

- administrator@Controller:~\$ openstack user create --domain default \
--password-prompt demo


```

administrator@Controller:~$ openstack user create --domain default \
> --password-prompt demo
User Password:
Repeat User Password:
+-----+-----+
| Field          | Value                               |
+-----+-----+
| domain_id     | default                             |
| enabled       | True                                |
| id            | dcfaddadac346f98151b45ecbc4ad7a    |
| name          | demo                                |
| options       | {}                                   |
| password_expires_at | None                                |
+-----+-----+
  
```

Δημιουργία user role

- administrator@Controller:~\$ openstack role create user

```

adminstrator@Controller:~$ openstack role create user
+-----+-----+
| Field          | Value                               |
+-----+-----+
| domain_id     | None                                |
| id            | 41a9bc2189d2497d9398d4db66c9e735   |
| name          | user                                |
+-----+-----+
  
```

4.8.5 Επαλήθευση λειτουργίας

Απενεργοποιούμε για λόγους ασφαλείας τον μηχανισμό προσωρινών κωδικών ταυτοποίησης. Παραμετροποίηση του `/etc/keystone/keystone-paste.ini` και αφαίρεση του `admin_token_auth` από τα `[pipeline:public_api]`, `[pipeline:admin_api]`, και `[pipeline:api_v3]`.

- root@Controller:~# vim /etc/keystone/keystone-paste.ini

```

[pipeline:public_api]
# The last item in this pipeline must be public_service or an equivalent
# application. It cannot be a filter.
pipeline = healthcheck cors stzelnit http_proxy_to_wsgi osprofiler url_normalize request_id admin_token_auth build_auth_context token_auth json_body ec2_extension public_service

[pipeline:admin_api]
# The last item in this pipeline must be admin_service or an equivalent
# application. It cannot be a filter.
pipeline = healthcheck cors stzelnit http_proxy_to_wsgi osprofiler url_normalize request_id admin_token_auth build_auth_context token_auth json_body ec2_extension s3_extension admin_service

[pipeline:api_v3]
# The last item in this pipeline must be service_v3 or an equivalent
# application. It cannot be a filter.
pipeline = healthcheck cors stzelnit http_proxy_to_wsgi osprofiler url_normalize request_id admin_token_auth build_auth_context token_auth json_body ec2_extension_v3 s3_extension service_v3
  
```

Κατάργηση της προσωρινής environment variable `OS_AUTH_URL` και `OS_PASSWORD`

- administrator@Controller:~\$ unset OS_AUTH_URL OS_PASSWORD

```

administrator@Controller:~$ unset OS_AUTH_URL OS_PASSWORD
administrator@Controller:~$ █
  
```

- ✓ Ως admin user, αιτούμαστε ένα authentication token
- administrator@Controller:~\$ openstack --os-auth-url http://controller:35357/v3 \
 --os-project-domain-name Default --os-user-domain-name Default \
 --os-project-name admin --os-username admin token issue
- ✓ Θα ζητηθεί το password του admin user


```

administrator@Controller:~$ openstack --os-auth-url http://controller:35357/v3 --os-project-domain-name Default --os-user-domain-name Default --os-project-name admin --os-username admin token issue
Password:
-----
| Field | Value |
-----
| expires | 2017-07-10T19:05:00+0000 |
| id | gAAAAB2V81x2V8dufB1cYbARl_rlv3gr_ZX0H0xk2J7wdr_3hGvV_TL9XP5vLoJdvlG6Bk1L_VFEVU5x70DPvAh2R009yI-Dor0W1ov0JLndH17E1cbs_tXNS0Noc47bnPnxQ6WUX2Gka9JPE-Pvn73-9P0nn20gBlw5bV5NsClld55xgfQJLo |
| project_id | 6686cfd5345a4486983f9f7bc776a8992 |
| user_id | 42926646ba9e44f78821989af39ac946 |
-----
administrator@Controller:~$

```

- administrator@Controller:~\$ openstack --os-auth-url http://controller:5000/v3 \
 --os-project-domain-name Default --os-user-domain-name Default \
 --os-project-name demo --os-username demo token issue

✓ Θα ζητηθεί το password του demo user

```

administrator@Controller:~$ openstack --os-auth-url http://controller:5000/v3 \
--os-project-domain-name Default --os-user-domain-name Default \
--os-project-name demo --os-username demo token issue
Password:
-----
| Field | Value |
-----
| expires | 2017-07-10T19:06:41+0000 |
| id | gAAAAB2V81x2V8dufB1cYbARl_rlv3gr_ZX0H0xk2J7wdr_3hGvV_TL9XP5vLoJdvlG6Bk1L_VFEVU5x70DPvAh2R009yI-Dor0W1ov0JLndH17E1cbs_tXNS0Noc47bnPnxQ6WUX2Gka9JPE-Pvn73-9P0nn20gBlw5bV5NsClld55xgfQJLo |
| project_id | 62c0827c3f094f45a81f9d9f726c4e |
| user_id | 6cf9d6d6ac346f91515145ecbc44d7a |
-----
administrator@Controller:~$

```

Δημιουργία OpenStack client environment scripts

Η προηγούμενη ενότητα χρησιμοποίησε έναν συνδυασμό environment variables και command options για να αλληλεπιδράσει με την υπηρεσία Ταυτότητα - Identity service μέσω του openstack client. Για να αυξήσει την αποδοτικότητα των λειτουργιών του client το OpenStack υποστηρίζει απλά scripts, γνωστά και ως αρχεία OpenRC. Αυτά τα scripts συνήθως περιέχουν κοινές επιλογές για όλους τους clients, αλλά υποστηρίζουν και μοναδικές επιλογές.

Δημιουργία των scripts

Παραμετροποίηση του αρχείου admin-openrc και προσθήκη του ακόλουθου περιεχομένου:

- administrator@Controller:~\$ sudo vim /admin-openrc.sh

```

export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=ADMIN_PASS
export OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2

```



```

export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=123456
export OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2

```

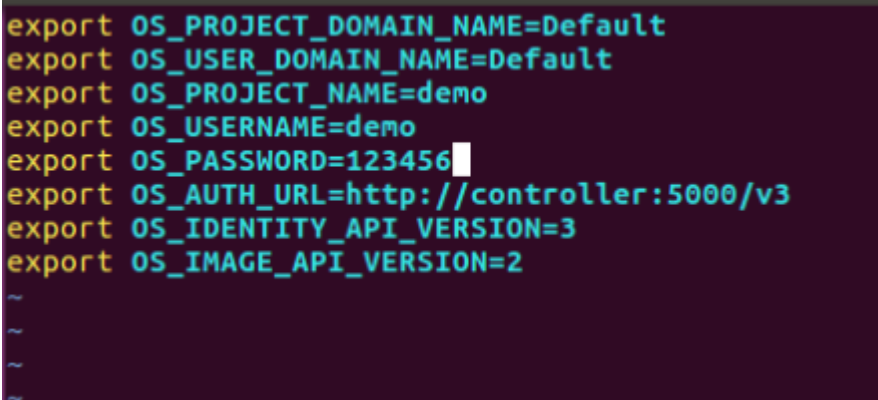
- ✓ Αντικατάσταση του **ADMIN_PASS** με το **password** του admin user που θέσαμε στην Υπηρεσία Ταυτότητας - Identity service

ADMIN_PASS: 123456

Παραμετροποίηση του αρχείου **demo-openrc** και προσθήκη του ακόλουθου περιεχομένου:

- administrator@Controller:~\$ sudo vim /demo-openrc.sh

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=DEMO_PASS
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```



```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=demo
export OS_USERNAME=demo
export OS_PASSWORD=123456
export OS_AUTH_URL=http://controller:5000/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

- ✓ Αντικατάσταση του **DEMO_PASS** με το **password** του demo user που θέσαμε στην Υπηρεσία Ταυτότητας - Identity service.

DEMO_PASS: 123456

Χρήση των Scripts

Φορτώνουμε το αρχείο admin-openrc για να γίνουν populate οι environment variables

- administrator@Controller:~\$. admin-openrc
- ✓ Αιτούμαστε για authentication token
- administrator@Controller:~\$ openstack token issue



```
administrator@Controller:~$ . admin-openrc
administrator@Controller:~$ openstack token issue
```

Field	Value
expires	2017-07-10T19:29:53+0000
id	gAAAAByBhC-S1KEK8DHmUVZ0Nqo81xk0aotabg88fLyFxx9zoh1c1Kbxyh-ECRrSL0EZ41Isrv57l00d0H09fp1y00PjHq4dLH0FPYscTtfsVUBpKRzotnGF8pg60ks5NXJ33BaR3ZfDwnJc0KZ7UmRUIL7bPC2L0vT3B_n7-J9
project_id	d080cfd345a440d0837df7bc70a08992
user_id	42926d60b9d6d4f78021909af790c946

```
administrator@Controller:~$
```

4.9 Image service

Η υπηρεσία Image (**glance**) επιτρέπει στους χρήστες να ανακαλύπτουν, να εγγράφονται και να ανακτούν virtual machine images.

Μπορεί κάποιος να αποθηκεύσει virtual machine images που διατίθενται μέσω του Image service σε διάφορες τοποθεσίες, από απλά συστήματα αρχείων μέχρι συστήματα αποθήκευσης αντικειμένων όπως το OpenStack Object Storage.

Σημαντικό είναι πριν προχωρήσουμε, να βεβαιωθούμε ότι ο Controller node έχει αρκετό διαθέσιμο χώρο στο directory `/var/lib/glance/images/`.

Το OpenStack Image service περιλαμβάνει τα ακόλουθα

glance-api

Αποδοχή Image API calls για ανάκτηση και αποθήκευση

glance-registry

Αποθηκεύει, επεξεργάζεται και ανακτά metadata σχετικά με images. Τα metadata περιλαμβάνουν στοιχεία όπως το μέγεθος και τον τύπο

Database

Αποθηκεύει τα image metadata και δίνει την δυνατότητα επιλογής βάσης δεδομένων που προτιμάμε.

✓ Οι περισσότερες εφαρμογές χρησιμοποιούν MySQL ή SQLite

Storage repository for image files

Υποστηρίζονται διάφοροι τύποι repository, συμπεριλαμβανομένων των κοινών file systems (ή οποιουδήποτε συστήματος αρχείων που είναι τοποθετημένος στον glance-api controller node), Object Storage, RADOS block devices, VMware datastore και HTTP

Metadata definition service

Ένα κοινό API για vendors, διαχειριστές, services, και τους χρήστες να καθορίζουν τα custom metadata. Αυτά τα metadata μπορούν να χρησιμοποιηθούν σε διαφορετικά resources, όπως images, artifacts, volumes, flavors, and aggregates (guide, n.d.)

4.9.1 Εγκατάσταση και διαμόρφωση

Προαπαιτούμενα

Δημιουργία Βάσης Δεδομένων – **glance**

➤ root@Controller:~# mysql -u root -p

```
MariaDB [(none)]> CREATE DATABASE glance;
```

```
DBDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' \
IDENTIFIED BY 'GLANCE_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' \
IDENTIFIED BY 'GLANCE_DBPASS';
```

```
MariaDB [(none)]> exit
```

```
root@Controller:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.0.29-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE glance;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
root@Controller:~#
```

GLANCE_DBPASS: 123456

Admin credentials

➤ root@Controller:~# . admin-openrc.sh

Service credentials – Δημιουργία

Glance user

➤ administrator@Controller:~\$ openstack user create --domain default \
--password-prompt glance

```
administrator@Controller:~$ openstack user create --domain default --password-prompt glance
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | default |
| enabled | True |
| id | e6b90641dc1843eb8221a515b1c4e9fe |
| name | glance |
| options | {} |
| password_expires_at | None |
+-----+-----+
```

Πρόσθεση ρόλου admin στον glance user και στο service project

➤ administrator@Controller:~\$ openstack role add --project service --user glance admin

Δημιουργία glance service

- administrator@Controller:~\$ openstack service create --name glance \
--description "OpenStack Image" image

```
administrator@Controller:~$ openstack role add --project service --user glance admin
administrator@Controller:~$ openstack service create --name glance \  
> --description "OpenStack Image" image
+-----+-----+
| Field | Value |
+-----+-----+
| description | OpenStack Image |
| enabled | True |
| id | 728dbe338fae4afcb030cb060103eb84 |
| name | glance |
| type | image |
+-----+-----+
administrator@Controller:~$
```

Δημιουργία Image service API endpoints

- administrator@Controller:~\$ openstack endpoint create --region RegionOne \
image public http://controller:9292

```
administrator@Controller:~$ openstack endpoint create --region RegionOne \  
> image public http://controller:9292
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 00f9090c4d87426392da902189127188 |
| interface | public |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 728dbe338fae4afcb030cb060103eb84 |
| service_name | glance |
| service_type | image |
| url | http://controller:9292 |
+-----+-----+
```

- administrator@Controller:~\$ openstack endpoint create --region RegionOne \
 image internal http://controller:9292

```

administrator@Controller:~$ openstack endpoint create --region RegionOne \  

> image internal http://controller:9292
+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | ca2020435abc4db297f9cd7de6e27cac       |
| interface  | internal                                 |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 728dbe338fae4afcb030cb060103eb84       |
| service_name | glance                                  |
| service_type | image                                   |
| url        | http://controller:9292                  |
+-----+-----+
  
```

- administrator@Controller:~\$ openstack endpoint create --region RegionOne \
 image admin http://controller:9292

```

administrator@Controller:~$ openstack endpoint create --region RegionOne \  

> image admin http://controller:9292
+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | bd240e188fe149629c9aa09914f6a5dc       |
| interface  | admin                                    |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | 728dbe338fae4afcb030cb060103eb84       |
| service_name | glance                                  |
| service_type | image                                   |
| url        | http://controller:9292                  |
+-----+-----+
  
```

Εγκατάσταση πακέτων

- root@Controller:~# apt install glance

Παραμετροποίηση του αρχείου /etc/glance/glance-api.conf

[database]

...

connection = mysql+pymysql://glance:GLANCE_DBPASS@controller/glance

GLANCE_DBPASS: 123456

```
[database]
#
# From oslo.db
#
# DEPRECATED: The file name to use with SQLite. (string value)
# Deprecated group/name - [DEFAULT]/sqlite_db
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Should use config option connection or slave_connection to connect the
# database.
#sqlite_db = oslo.sqlite
sqlite_db = /var/lib/glance/glance.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecated group/name - [DEFAULT]/sqlite_synchronous
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
backend = sqlalchemy

# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://glance:123456@controller/glance
```

[keystone_authtoken]

...

```
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = GLANCE_PASS
```

GLANCE_PASS: 123456


```
[keystone_authtoken]
#
# From keystonemiddleware.auth_token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users. Unauthenticated
# clients are redirected to this endpoint to authenticate. Although this
# endpoint should ideally be unversioned, client support in the wild varies.
# If you're using a versioned v2 endpoint here, then this should *not* be the
# same endpoint the service user utilizes for validating tokens, because normal
# end users may not be able to reach that endpoint. (string value)
#auth_uri = <None>
auth_uri = http://controller:5000
auth_uri = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = 123456
```

[paste_deploy]

...

flavor = keystone

```
[paste_deploy]
#
# From glance.api
#
#
# Deployment flavor to use in the server application pipeline.
#
# Provide a string value representing the appropriate deployment
# flavor used in the server application pipeline. This is typically
# the partial name of a pipeline in the paste configuration file with
# the service name removed.
#
# For example, if your paste section name in the paste configuration
# file is [pipeline:glance-api-keystone], set ``flavor`` to
# ``keystone``.
#
# Possible values:
# * String value representing a partial pipeline name.
#
# Related Options:
# * config_file
#
# (string value)
flavor = keystone
```

```
[glance_store]
...
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
```

```
[glance_store]
#
# From glance.store
#
#
# List of enabled Glance stores.
#
# Register the storage backends to use for storing disk images
# as a comma separated list. The default stores enabled for
# storing disk images with Glance are ``file`` and ``http``.
#
# Possible values:
#   * A comma separated list that could include:
#     * file
#     * http
#     * swift
#     * rbd
#     * sheepdog
#     * cinder
#     * vmware
#
# Related Options:
#   * default_store
#
# (list value)
#stores = file,http
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
#
```

Παραμετροποίηση του αρχείου `/etc/glance/glance-registry.conf`

```
[database]
...
connection = mysql+pymysql://glance:GLANCE_DBPASS@controller/glance
```

GLANCE_DBPASS: 123456

```
[database]
#
# From oslo.db
#
# DEPRECATED: The file name to use with SQLite. (string value)
# Deprecatd group/name - [DEFAULT]/sqlite_db
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Should use config option connection or slave_connection to connect the
# database.
#sqlite_db = oslo.sqlite
sqlite_db = /var/lib/glance/glance.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecatd group/name - [DEFAULT]/sqlite_synchronous
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecatd group/name - [DEFAULT]/db_backend
backend = sqlalchemy

# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecatd group/name - [DEFAULT]/sql_connection
# Deprecatd group/name - [DATABASE]/sql_connection
# Deprecatd group/name - [sql]/connection
connection = mysql+pymysql://glance:123456@controller/glance
```

[keystone_authtoken]

...

```
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = GLANCE_PASS
```

```
[keystone_authtoken]
#
# From keystonemiddleware.auth_token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users. Unauthenticated
# clients are redirected to this endpoint to authenticate. Although this
# endpoint should ideally be unversioned, client support in the wild varies.
# If you're using a versioned v2 endpoint here, then this should *not* be the
# same endpoint the service user utilizes for validating tokens, because normal
# end users may not be able to reach that endpoint. (string value)
#auth_uri = <None>
auth_url = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = glance
password = 123456
# API version of the admin Identity API endpoint. (string value)
#auth_version = <None>
```

GLANCE_PASS: 123456

[paste_deploy]

...

flavor = keystone

```
[paste_deploy]
#
# From glance.registry
#
#
# Deployment flavor to use in the server application pipeline.
#
# Provide a string value representing the appropriate deployment
# flavor used in the server application pipeline. This is typically
# the partial name of a pipeline in the paste configuration file with
# the service name removed.
#
# For example, if your paste section name in the paste configuration
# file is [pipeline:glance-api-keystone], set ``flavor`` to
# ``keystone``.
#
# Possible values:
# * String value representing a partial pipeline name.
#
# Related Options:
# * config_file
#
# (string value)
flavor = keystone
```

Populate the Image service database

➤ root@Controller:~# su -s /bin/sh -c "glance-manage db_sync" glance

```
root@Controller:~# su -s /bin/sh -c "glance-manage db_sync" glance
/usr/lib/python2.7/dist-packages/oslo_db/sqlalchemy/enginefacade.py:1241: OsloDBDeprecationWarning: EngineFacade is deprecated; please use oslo_db.sqlalchemy.enginefacade
expire_on_commit=expire_on_commit, _conf=conf)
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
INFO [alembic.runtime.migration] Running upgrade -> liberty, liberty initial
INFO [alembic.runtime.migration] Running upgrade liberty -> mitaka01, add index on created_at and updated_at columns of 'images' table
INFO [alembic.runtime.migration] Running upgrade mitaka01 -> mitaka02, update metadef os_nova_server
INFO [alembic.runtime.migration] Running upgrade mitaka02 -> ocata01, add visibility to and remove is_public from images
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
Upgraded database to: ocata01, current revision(s): ocata01
root@Controller:~#
```

✓ Επανεκκίνηση Image services

➤ root@Controller:~# service glance-registry restart

➤ root@Controller:~# service glance-api restart

4.9.2 Επαλήθευση λειτουργίας

➤ administrator@Controller:~\$. admin-openrc.sh

Download the source image

➤ administrator@Controller:~\$ wget http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img

```

root@Controller:~# admin-openrc.sh
root@Controller:~# exit
administrator@Controller:~$ wget http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img
--2017-07-14 01:08:51-- http://download.cirros-cloud.net/0.3.4/cirros-0.3.4-x86_64-disk.img
Resolving download.cirros-cloud.net (download.cirros-cloud.net)... 64.90.42.85, 2007:f298:1:a036::bdc:a72a
Connecting to download.cirros-cloud.net (download.cirros-cloud.net)|64.90.42.85|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13287936 (13M) [text/plain]
Saving to: 'cirros-0.3.4-x86_64-disk.img'
cirros-0.3.4-x86_64-disk.img 100%[=====] 12.67M 2.76MB/s in 0.1s
2017-07-14 01:08:59 (1.56 MB/s) - 'cirros-0.3.4-x86_64-disk.img' saved [13287936/13287936]

```

Μεταφόρτωση του image στο Image service χρησιμοποιώντας τη QCOW2 disk format, bare container format και public visibility, ώστε να έχουν πρόσβαση σε όλα τα έργα

- administrator@Controller:~\$ openstack image create "cirros" \
 --file cirros-0.3.4-x86_64-disk.img \
 --disk-format qcow2 --container-format bare \
 --public

```

administrator@Controller:~$ openstack image create "cirros" \
> --file cirros-0.3.4-x86_64-disk.img \
> --disk-format qcow2 --container-format bare \
> --public
+-----+
| Field | Value |
+-----+
| checksum | ee1eca47dc88f4879d8a229cc70a07c6 |
| container_format | bare |
| created_at | 2017-07-13T22:09:45Z |
| disk_format | qcow2 |
| file | /v2/images/626b13af-f75b-4e79-a9f4-0d7b201cb435/file |
| id | 626b13af-f75b-4e79-a9f4-0d7b201cb435 |
| min_disk | 0 |
| min_ram | 0 |
| name | cirros |
| owner | d680cfd5345a4406983fdf7bc70a8992 |
| protected | False |
| schema | /v2/schemas/image |
| size | 13287936 |
| status | active |
| tags | |
| updated_at | 2017-07-13T22:09:45Z |
| virtual_size | None |
| visibility | public |
+-----+

```

Επιβεβαίωση μεταφόρτωσης του image και επικύρωση attributes

- administrator@Controller:~\$ openstack image list

```

administrator@Controller:~$ openstack image list
+-----+-----+-----+
| ID | Name | Status |
+-----+-----+-----+
| 626b13af-f75b-4e79-a9f4-0d7b201cb435 | cirros | active |
+-----+-----+-----+
administrator@Controller:~$ █

```

4.10 Compute service

- Το OpenStack Compute χρησιμοποιείτε για να φιλοξενήσουμε και να διαχειριστούμε τα συστήματα υπολογιστικού νέφους - cloud computing systems
- Είναι ένα σημαντικό μέρος ενός συστήματος Infrastructure-as-a-Service (IaaS) και οι κύριες ενότητες υλοποιούνται σε Python
- Αλληλεπιδρά με το OpenStack Identity για έλεγχο ταυτότητας
- Το OpenStack Image service για images δίσκων και διακομιστών
- Το OpenStack dashboard για το χρήστη και το διαχειριστικό interface

- Η πρόσβαση στο image περιορίζεται από τα projects και τους χρήστες
- Τα quotas είναι περιορισμένα ανά έργο

4.10.1 Το OpenStack Compute αποτελείται από τα ακόλουθα

- nova-api **service**
- nova-api-metadata **service**
- nova-compute **service**
- nova-placement-api **service****
- nova-scheduler **service**
- nova-conductor **module**
- nova-cert **module**
- nova-network worker **daemon**
- nova-consoleauth **daemon**
- nova-novncproxy **daemon**
- nova-spicehtml5proxy **daemon**
- nova-xvncproxy **daemon**
- nova-cert **daemon**
- nova **client**
- The queue
- SQL database

4.10.2 Εγκατάσταση και διαμόρφωση – Controller node

Προαπαιτούμενα

Δημιουργία Βάσεις Δεδομένων – nova_api & nova

➤ root@Controller:~# mysql -u root -p

```
MariaDB [(none)]> CREATE DATABASE nova_api;
```

```
MariaDB [(none)]> CREATE DATABASE nova;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' \
IDENTIFIED BY 'NOVA_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%' \
IDENTIFIED BY 'NOVA_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' \
IDENTIFIED BY 'NOVA_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' \
IDENTIFIED BY 'NOVA_DBPASS';
```

```
MariaDB [(none)]> exit
```

NOVA_DBPASS: **123456**


```

root@Controller:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.0.29-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE nova_api;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> CREATE DATABASE nova;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'localhost' \
-> IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.28 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova_api.* TO 'nova'@'%' \
-> IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' \
-> IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' \
-> IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
  
```

Admin credentials

➤ root@Controller:~# . admin-openrc.sh

Service credentials – Δημιουργία

Nova user

➤ administrator@Controller:~\$ openstack user create --domain default \
--password-prompt nova

Πρόσθεση ρόλου admin στον nova user και στο service project

➤ administrator@Controller:~\$ openstack role add --project service --user nova admin

```

root@Controller:~# . admin-openrc.sh
root@Controller:~# openstack user create --domain default \
> --password-prompt nova
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | default |
| enabled | True |
| id | 03ebb164472f45acb4bb9485854eb9d5 |
| name | nova |
| options | {} |
| password_expires_at | None |
+-----+-----+
root@Controller:~# openstack role add --project service --user nova admin
root@Controller:~# █
  
```


Δημιουργία Nova service

- root@Controller:~# openstack service create --name nova --description \ "OpenStack Compute" compute

```
root@Controller:~# openstack role add --project service --user nova admin
root@Controller:~# openstack service create --name nova \
> --description "OpenStack Compute" compute
```

Field	Value
description	OpenStack Compute
enabled	True
id	870bc1468f104804bbac25dfd910ecdd
name	nova
type	compute

```
root@Controller:~#
```

Δημιουργία Compute service API endpoints

- root@Controller:~# openstack endpoint create --region RegionOne \ compute public http://controller:8774/v2.1/%(tenant_id)s

```
root@Controller:~# openstack endpoint create --region RegionOne \ compute public http://controller:8774/v2.1/%(tenant_id)s
```

Field	Value
enabled	True
id	6e30524552394db4bb820b3392d93faf
interface	public
region	RegionOne
region_id	RegionOne
service_id	870bc1468f104804bbac25dfd910ecdd
service_name	nova
service_type	compute
url	http://controller:8774/v2.1/%(tenant_id)s

- root@Controller:~# openstack endpoint create --region RegionOne \ compute internal http://controller:8774/v2.1/%(tenant_id)s

```
root@Controller:~# openstack endpoint create --region RegionOne \
> compute internal http://controller:8774/v2.1/%(tenant_id)s
```

Field	Value
enabled	True
id	fd94027626df488eafb228ed482e7b22
interface	internal
region	RegionOne
region_id	RegionOne
service_id	870bc1468f104804bbac25dfd910ecdd
service_name	nova
service_type	compute
url	http://controller:8774/v2.1/%(tenant_id)s

- root@Controller:~# openstack endpoint create --region RegionOne \ compute admin http://controller:8774/v2.1/%(tenant_id)s

```
root@Controller:~# openstack endpoint create --region RegionOne \
> compute admin http://controller:8774/v2.1/%(tenant_id)s
```

Field	Value
enabled	True
id	899f8e0a1ac846bd9c41dfac58ec30c8
interface	admin
region	RegionOne
region_id	RegionOne
service_id	870bc1468f104804bbac25dfd910ecdd
service_name	nova
service_type	compute
url	http://controller:8774/v2.1/%(tenant_id)s

Δημιουργία Placement service User που χρησιμοποιεί το επιλεγμένο PLACEMENT_PASS

- root@Controller:~# openstack user create --domain default --password-prompt placement
User Password:

Πρόσθεση του Placement user στο service project με admin role

- root@Controller:~# openstack role add --project service --user placement admin

Δημιουργία Placement API entry in the service catalog

- root@Controller:~# openstack service create --name placement --description \"Placement API\" placement

Δημιουργία Placement API service endpoints

- root@Controller:~# openstack endpoint create --region RegionOne placement \ public http://controller:8778
- root@Controller:~# openstack endpoint create --region RegionOne placement \ internal http://controller:8778
- root@Controller:~# openstack endpoint create --region RegionOne placement \ admin http://controller:8778

Εγκατάσταση πακέτων

- root@Controller:~# apt install nova-api nova-conductor nova-consoleauth \ nova-novncproxy nova-scheduler nova-placement-api

Παραμετροποίηση του αρχείου /etc/nova/nova.conf

[api_database]

...

connection = mysql+pymysql://nova:NOVA_DBPASS@controller/nova_api

```
[api_database]
#
# The *Nova API Database* is a separate database which is used for information
# which is used across *cells*. This database is mandatory since the Mitaka
# release (13.0.0).
#
# From nova.conf
#
# The SQLAlchemy connection string to use to connect to the database. (string
# value)
#connection=sqlite:///var/lib/nova/nova.sqlite
connection = mysql+pymysql://nova:123456@controller/nova_api
#
# If True, SQLite uses synchronous mode. (boolean value)
#sqlite_synchronous=true
```

[database]

...

connection = mysql+pymysql://nova:NOVA_DBPASS@controller/nova

```
# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend=sqlalchemy

# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://nova:123456@controller/nova
#
# The SQLAlchemy connection string to use to connect to the slave database.
# (string value)
#slave_connection=<None>
```

NOVA_PASS: 123456

[DEFAULT]

...

transport_url = rabbit://openstack:RABBIT_PASS@controller

```
# Size of executor thread pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_thread_pool_size
#executor_thread_pool_size=64

# Seconds to wait for a response from a call. (integer value)
#rpc_response_timeout=60

# A URL representing the messaging driver to use and its full configuration.
# (string value)
transport_url= rabbit://openstack:123456@controller
#
# DEPRECATED: The messaging driver to use, defaults to rabbit. Other drivers
# include amqp and zmq. (string value)
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Replaced by [DEFAULT]/transport_url
#rpc_backend=rabbit
```

RABBIT_PASS: 123456

[DEFAULT]

...

auth_strategy = keystone

```
[DEFAULT]i
#
# From nova.conf
#
# DEPRECATED:
# When returning instance metadata, this is the class that is used
# for getting vendor metadata when that class isn't specified in the individual
# request. The value should be the full dot-separated path to the class to use.
#
# Possible values:
# * Any valid dot-separated class path that can be imported.
# (string value)
# This option is deprecated for removal since 13.0.0.
# Its value may be silently ignored in the future.
#vendordata_driver=nova.api.metadata.vendordata_json.JsonFileVendorData

auth_strategy = keystone
#
# DEPRECATED:
# This option is used to enable or disable quota checking for tenant networks.
#
```

[keystone_authtoken]

...

auth_uri = http://controller:5000

auth_url = http://controller:35357

```
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = NOVA_PASS
```

```
[keystone_auth_token]
#
# From keystonemiddleware.auth_token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users. Unauthenticated
# clients are redirected to this endpoint to authenticate. Although this
# endpoint should ideally be unversioned, client support in the wild varies.
# If you're using a versioned v2 endpoint here, then this should *not* be the
# same endpoint the service user utilizes for validating tokens, because normal
# end users may not be able to reach that endpoint. (string value)
#auth_url=<None>

auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = 123456
```

NOVA_PASS: 123456

[DEFAULT]

...

my_ip = 172.16.10.231

```
[DEFAULT]
#
# From nova.conf
#
my_ip = 172.16.10.231
```

[DEFAULT]

...

use_neutron = True

firewall_driver = nova.virt.firewall.NoopFirewallDriver

```
# Related options:
#
# * ``use_neutron``: This must be set to ``False`` to enable ``nova-network``
# networking
# (string value)
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

[vnc]

...

vncserver_listen = \$my_ip

vncserver_proxyclient_address = \$my_ip

```
[vnc]
#
# Virtual Network Computer (VNC) can be used to provide remote desktop
# console access to instances for tenants and/or administrators.
#
# From nova.conf
#
#
# Enable VNC related features.
#
# Guests will get created with graphical devices to support this. Clients
# (for example Horizon) can then establish a VNC connection to the guest.
# (boolean value)
# Deprecated group/name - [DEFAULT]/vnc_enabled
#enabled=true

vncserver_listen = $my_ip
vncserver_proxyclient_address = $my_ip
```

[glance]

...

api_servers = http://controller:9292

```
[glance]
# Configuration options for the Image service
#
# From nova.conf
#
#
# List of glance api servers endpoints available to nova.
# https is used for ssl-based glance api servers.
#
# Possible values:
#
# * A list of any fully qualified url of the form
# "scheme://hostname:port[/path]"
# (i.e. "http://10.0.1.0:9292" or "https://my.glance.server/image").
# (list value)
api_servers = http://controller:9292
```

[oslo_concurrency]

...

lock_path = /var/lib/nova/tmp

```
[oslo_concurrency]
#
# From oslo.concurrency
#
#
# Enables or disables inter-process locks. (boolean value)
# Deprecated group/name - [DEFAULT]/disable_process_locking
#disable_process_locking=false
#
# Directory to use for lock files. For security, the specified directory should
# only be writable by the user running the processes that need locking. Defaults
# to environment variable OSLO_LOCK_PATH. If OSLO_LOCK_PATH is not set in the
# environment, use the Python tempfile.gettempdir function to find a suitable
# location. If external locks are used, a lock path must be set. (string value)
# Deprecated group/name - [DEFAULT]/lock_path
#lock_path=/var/lock/nova
lock_path = /var/lib/nova/tmp
```

✓ Διαγραφή `log-dir` από την ενότητα [DEFAULT]

[placement]

...

```
os_region_name = RegionOne
project_domain_name = default
project_name = service
auth_type = password
user_domain_name = default
auth_url = http://controller:35357/v3
username = placement
password = PLACEMENT_PASS
```

```
[placement]
# (string value)
os_region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:35357/v3
username = placement
password = 123456
```

PLACEMENT_PASS: **123456**

Populate the Compute databases

- root@Controller:~# su -s /bin/sh -c "nova-manage api_db sync" nova
- root@Controller:~# su -s /bin/sh -c "nova-manage db sync" nova

Register the cell0 database

- root@Controller:~# su -s /bin/sh -c "nova-manage cell_v2 map_cell0" nova

Δημιουργία cell1 cell

- root@Controller:~# su -s /bin/sh -c "nova-manage cell_v2 create_cell --name=cell1 \ --verbose" nova

Επαλήθευση nova cell0 και cell1

- root@Controller:~# nova-manage cell_v2 list_cells
- ✓ Επανεκκίνηση του Compute service
- root@Controller:~ # service nova-api restart
- root@Controller:~ # service nova-consoleauth restart
- root@Controller:~ # service nova-scheduler restart
- root@Controller:~ # service nova-conductor restart
- root@Controller:~ # service nova-novncproxy restart

4.10. 3 Εγκατάσταση και διαμόρφωση – Compute node

Εγκατάσταση πακέτων

➤ root@Compute:~# apt install nova-compute

Παραμετροποίηση του αρχείου /etc/nova/nova.conf

[DEFAULT]

...

transport_url = rabbit://openstack:RABBIT_PASS@controller

RABBIT_DBPASS: 123456

```
# Size of executor thread pool. (integer value)
# Deprecated group/name - [DEFAULT]/rpc_thread_pool_size
#executor_thread_pool_size=64

# Seconds to wait for a response from a call. (integer value)
#rpc_response_timeout=60

# A URL representing the messaging driver to use and its full configuration.
# (string value)
transport_url = rabbit://openstack:123456@controller

# DEPRECATED: The messaging driver to use, defaults to rabbit. Other drivers
# include amqp and zmq. (string value)
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Replaced by [DEFAULT]/transport_url
#rpc_backend=rabbit
```

[DEFAULT]

...

auth_strategy = keystone

```
[DEFAULT]

#
# From nova.conf
#

# DEPRECATED:
# When returning instance metadata, this is the class that is used
# for getting vendor metadata when that class isn't specified in the individual
# request. The value should be the full dot-separated path to the class to use.
#
# Possible values:
#
# * Any valid dot-separated class path that can be imported.
# (string value)
# This option is deprecated for removal since 13.0.0.
# Its value may be silently ignored in the future.
#vendordata_driver=nova.api.metadata.vendordata_json.JsonFileVendorData

auth_strategy = keystone
```

[keystone_authtoken]

...


```
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = NOVA_PASS
```

NOVA_DBPASS: **123456**

```
[keystone_authtoken]
#
# From keystonemiddleware.auth_token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users. Unauthenticated
# clients are redirected to this endpoint to authenticate. Although this
# endpoint should ideally be unversioned, client support in the wild varies.
# If you're using a versioned v2 endpoint here, then this should *not* be the
# same endpoint the service user utilizes for validating tokens, because normal
# end users may not be able to reach that endpoint. (string value)
#auth_uri=<None>
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = nova
password = 123456
```

[DEFAULT]

...

```
my_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
```

MANAGEMENT_INTERFACE_IP_ADDRESS: **172.16.10.232**

```
# * vpn_ip
# (string value)
my_ip=172.16.10.232
#
```

[DEFAULT]

...

```
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
```

✓ Απενεργοποίηση του firewall nova.virt.firewall.NoopFirewallDriver firewall driver.

```
# DEPRECATED: Whether to use Neutron or Nova Network as the back end for
# networking. Defaults to False (indicating Nova network).Set to True to use
# neutron. (boolean value)
# This option is deprecated for removal since 15.0.0.
# Its value may be silently ignored in the future.
# Reason:
# nova-network is deprecated, as are any related configuration options.
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver
#
```

[vnc]

...

enabled = True

vncserver_listen = 0.0.0.0

vncserver_proxyclient_address = \$my_ip

novncproxy_base_url = http://controller:6080/vnc_auto.html

```
[vnc]
#
# Virtual Network Computer (VNC) can be used to provide remote desktop
# console access to instances for tenants and/or administrators.
#
# From nova.conf
#
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://controller:6080/vnc_auto.html
#
```

[glance]

...

api_servers = http://controller:9292

```
[glance]
# Configuration options for the Image service
#
# From nova.conf
#
#
# List of glance api servers endpoints available to nova.
# https is used for ssl-based glance api servers.
# Possible values:
#
# * A list of any fully qualified url of the form
# "scheme://hostname:port[/path]"
# (i.e. "http://10.0.1.0:9292" or "https://my.glance.server/image").
# (list value)
api_servers = http://controller:9292
```

[oslo_concurrency]

...

lock_path = /var/lib/nova/tmp

```
[oslo_concurrency]
#
# From oslo.concurrency
#
# Enables or disables inter-process locks. (boolean value)
# Deprecated group/name - [DEFAULT]/disable_process_locking
#disable_process_locking=false
#
# Directory to use for lock files. For security, the specified directory should
# only be writable by the user running the processes that need locking. Defaults
# to environment variable OSLO_LOCK_PATH. If OSLO_LOCK_PATH is not set in the
# environment, use the Python tempfile.gettempdir function to find a suitable
# location. If external locks are used, a lock path must be set. (string value)
# Deprecated group/name - [DEFAULT]/lock_path
lock_path = /var/lib/nova/tmp
```

[placement]

...

```
os_region_name = RegionOne
project_domain_name = default
project_name = service
auth_type = password
user_domain_name = default
auth_url = http://controller:35357/v3
username = placement
password = PLACEMENT_PASS
```

```
# (string value)
os_region_name = RegionOne
project_domain_name = default
project_name = service
auth_type = password
user_domain_name = default
auth_url = http://controller:35357/v3
username = placement
password = 123456
```

PLACEMENT_PASS: **123456**

✓ Διαγραφή `log-dir` από την ενότητα `[DEFAULT]`

Discover compute hosts

➤ `root@Controller:~# su -s /bin/sh -c "nova-manage cell_v2 discover_hosts --verbose" nova`

Όταν προστεθούν νέοι compute nodes, πρέπει να εκτελεστεί `"nova-manage cell_v2 discover_hosts"` στον Controller node για να τους καταχωρίσει. Εναλλακτικά, μπορούμε να ορίσουμε ένα κατάλληλο διάστημα.

➤ `root@Controller:~# vim /etc/nova/nova.conf:`

[scheduler]

```
discover_hosts_in_cells_interval = 300
```

4.10.4 Ολοκλήρωση εγκατάστασης

Hardware acceleration for virtual machines

- root@Compute:~# egrep -c '(vmx|svm)' /proc/cpuinfo

```
root@Compute:~# egrep -c '(vmx|svm)' /proc/cpuinfo
2
root@Compute:~#
```

Αν αυτή η εντολή μας επιστρέψει τιμή ίση ή μεγαλύτερη του ένα (1) τότε ο Compute node υποστηρίζει hardware acceleration και δεν απαιτεί πρόσθετη διαμόρφωση, ενώ αντίθετα με την τιμή μικρότερη του ένα (1) ο Compute node δεν υποστηρίζει hardware acceleration και πρέπει να ρυθμιστεί το libvirt να χρησιμοποιεί QEMU αντί KVM.

- root@Compute:~# vim /etc/nova/nova-compute.conf

[libvirt]

...

virt_type = kvm

```
[DEFAULT]
compute_driver=libvirt.LibvirtDriver
[libvirt]
virt_type=kvm
```

- ✓ Επανεκκίνηση του nova-compute service

- root@Compute:~# service nova-compute restart

4.10.5 Επαλήθευση λειτουργίας – Controller node

Admin credentials

- root@Controller:~# . admin-openrc.sh

Επαλήθευση της επιτυχούς έναρξης και καταχώρησης κάθε διαδικασίας

- root@Controller:~# openstack compute service list

```
root@Controller:~# . admin-openrc.sh
root@Controller:~# openstack compute service list
+-----+-----+-----+-----+-----+-----+-----+
| ID | Binary          | Host       | Zone   | Status | State | Updated At          |
+-----+-----+-----+-----+-----+-----+-----+
| 6  | nova-consoleauth | Controller | internal | enabled | up    | 2017-07-17T17:48:50.000000 |
| 7  | nova-scheduler   | Controller | internal | enabled | up    | 2017-07-17T17:48:50.000000 |
| 8  | nova-conductor   | Controller | internal | enabled | up    | 2017-07-17T17:48:41.000000 |
| 9  | nova-compute     | Compute   | nova   | enabled | up    | 2017-07-17T17:48:47.000000 |
+-----+-----+-----+-----+-----+-----+-----+
root@Controller:~#
```

4.11 Networking service

Αυτό το κεφάλαιο εξηγεί τον τρόπο εγκατάστασης και ρύθμισης παραμέτρων του Networking service (neutron) χρησιμοποιώντας τα provider networks ή τα self-service networks.

Το OpenStack Networking (neutron) επιτρέπει να δημιουργήσουμε και να συνδέσουμε συσκευές διασύνδεσης που διαχειρίζονται άλλες υπηρεσίες OpenStack σε δίκτυα. Τα Plug-ins μπορούν να υλοποιηθούν για να φιλοξενήσουν διαφορετικό εξοπλισμό δικτύωσης και λογισμικού, παρέχοντας ευελιξία στην αρχιτεκτονική του OpenStack και στην ανάπτυξη.

Περιλαμβάνει τα ακόλουθα στοιχεία:

- neutron-server

Αποδέχεται και δρομολογεί αιτήματα API στο κατάλληλο OpenStack Networking plug-in.

- OpenStack Networking plug-ins and agents

Μας δίνει την δυνατότητα να συνδέουμε και να αποσυνδέουμε τις πόρτες (plug & unplug ports) τις πόρτες, να δημιουργήσουμε δίκτυα ή υποδίκτυα και να παρέχουμε διευθύνσεις IP. Αυτά τα plug-ins και οι agents διαφέρουν ανάλογα με τον βέντορα και τις τεχνολογίες που χρησιμοποιούνται στο κάθε cloud.

- Messaging queue

Χρησιμοποιείται από τις περισσότερες εγκαταστάσεις του OpenStack Networking για τη δρομολόγηση πληροφοριών μεταξύ του neutron-server και διαφόρων agents. Λειτουργεί επίσης ως βάση δεδομένων για την αποθήκευση της κατάστασης δικτύωσης για συγκεκριμένες plug-ins. (guide, n.d.)

4.11.1 Εγκατάσταση και διαμόρφωση Controller node

Προαπαιτούμενα

Δημιουργία Βάσης Δεδομένων – **neutron**

- root@Controller:~# mysql -u root -p

```
MariaDB [(none)]> CREATE DATABASE neutron;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' \
IDENTIFIED BY 'NEUTRON_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'% ' \
IDENTIFIED BY 'NEUTRON_DBPASS';
```

```
MariaDB [(none)]> exit
```

```

root@Controller:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 100
Server version: 10.0.29-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE neutron;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
root@Controller:~#
  
```

NEUTRON_DBPASS: 123456

Admin credentials

➤ root@Controller:~# . admin-openrc.sh

Δημιουργία neutron user

➤ root@Controller:~# openstack user create --domain default --password-prompt neutron

```

root@Controller:~# . admin-openrc.sh
root@Controller:~# openstack user create --domain default --password-prompt neutron
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | default |
| enabled | True |
| id | fdcc4749846e4b088bc6d031ae39a606 |
| name | neutron |
| options | {} |
| password_expires_at | None |
+-----+-----+
root@Controller:~# openstack role add --project service --user neutron admin
root@Controller:~#
  
```

Πρόσθεση admin role στον neutron user

➤ root@Controller:~# openstack role add --project service --user neutron admin

Δημιουργία neutron service

➤ root@Controller:~# openstack service create --name neutron \
 --description "OpenStack Networking" network

```

root@Controller:~# openstack service create --name neutron --description "OpenStack Networking" network
+-----+-----+
| Field | Value |
+-----+-----+
| description | OpenStack Networking |
| enabled | True |
| id | 94c302b803b74d93aa1f2e0dd3ad6acd |
| name | neutron |
| type | network |
+-----+-----+
root@Controller:~#
  
```

Δημιουργία Networking service API endpoints

- root@Controller:~# openstack endpoint create --region RegionOne \
network public http://controller:9696

```
root@Controller:~# openstack endpoint create --region RegionOne network public http://controller:9696
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 776b2df111a34337a051058b83a931de |
| interface | public |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 94c302b803b74d93aa1f2e0dd3ad6acd |
| service_name | neutron |
| service_type | network |
| url | http://controller:9696 |
+-----+-----+
```

- root@Controller:~# openstack endpoint create --region RegionOne \
network internal http://controller:9696

```
root@Controller:~# openstack endpoint create --region RegionOne network internal http://controller:9696
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | f06396f7f32f4be7b63a5890a2104b69 |
| interface | internal |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 94c302b803b74d93aa1f2e0dd3ad6acd |
| service_name | neutron |
| service_type | network |
| url | http://controller:9696 |
+-----+-----+
```

- root@Controller:~# openstack endpoint create --region RegionOne \
network admin http://controller:9696

```
root@Controller:~# openstack endpoint create --region RegionOne network admin http://controller:9696
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 7281ec3cdf141029b3ee0c6d79e0d1d |
| interface | admin |
| region | RegionOne |
| region_id | RegionOne |
| service_id | 94c302b803b74d93aa1f2e0dd3ad6acd |
| service_name | neutron |
| service_type | network |
| url | http://controller:9696 |
+-----+-----+
root@Controller:~# █
```

4.11.2 Configure networking options

Self-service networks

Εγκατάσταση στοιχείων

- root@Controller:~# apt install neutron-server neutron-plugin-ml2 \
neutron-linuxbridge-agent neutron-l3-agent neutron-dhcp-agent \
neutron-metadata-agent

Διαμόρφωση server component

Παραμετροποίηση του αρχείου /etc/neutron/neutron.conf

➤ root@Controller:~# vim /etc/neutron/neutron.conf

[database]

...

connection = mysql+pymysql://neutron:NEUTRON_DBPASS@controller/neutron

```
[database]
#
# From neutron.db
#
# Database engine for which script will be generated when using offline
# migration. (string value)
#engine =

#
# From oslo.db
#
# DEPRECATED: The file name to use with SQLite. (string value)
# Deprecated group/name - [DEFAULT]/sqlite_db
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Should use config option connection or slave_connection to connect
# the database.
#sqlite_db = oslo.sqlite

# If True, SQLite uses synchronous mode. (boolean value)
# Deprecated group/name - [DEFAULT]/sqlite_synchronous
#sqlite_synchronous = true

# The back end to use for the database. (string value)
# Deprecated group/name - [DEFAULT]/db_backend
#backend = sqlalchemy

# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
#connection = sqlite:///var/lib/neutron/neutron.sqlite
connection = mysql+pymysql://neutron:123456@controller/neutron
```

NEUTRON_DBPASS: 123456

[DEFAULT]

...

core_plugin = ml2

service_plugins = router

allow_overlapping_ips = True

```
[DEFAULT]

#
# From neutron
#
# Where to store Neutron state files. This directory must be writable by the
# agent. (string value)
#state_path = /var/lib/neutron

# The host IP to bind to (string value)
#bind_host = 0.0.0.0

# The port to bind to (port value)
# Minimum value: 0
# Maximum value: 65535
#bind_port = 9696

# The path for API extensions. Note that this can be a colon-separated list of
# paths. For example: api_extensions_path =
# extensions:/path/to/more/exts:/even/more/exts. The __path__ of
# neutron.extensions is appended to this, so if your extensions are in there
# you don't need to specify them here. (string value)
#api_extensions_path =

# The type of authentication to use (string value)
auth_strategy = keystone

# The core plugin Neutron will use (string value)
core_plugin = ml2

# The service plugins Neutron will use (list value)
service_plugins = router

# Allow overlapping IP support in Neutron. Attention: the following parameter
# MUST be set to False if Neutron is being used in conjunction with Nova
# security groups. (boolean value)
allow_overlapping_ips = True
```

[DEFAULT]

...

transport_url = rabbit://openstack:RABBIT_PASS@controller

```
# A URL representing the messaging driver to use and its full configuration.
# (string value)
transport_url = rabbit://openstack:123456@controller
```

RABBIT_PASS: 123456

[DEFAULT]

...

auth_strategy = keystone

```
[DEFAULT]

#
# From neutron
#

# Where to store Neutron state files. This directory must be writable by the
# agent. (string value)
#state_path = /var/lib/neutron

# The host IP to bind to (string value)
#bind_host = 0.0.0.0

# The port to bind to (port value)
# Minimum value: 0
# Maximum value: 65535
#bind_port = 9696

# The path for API extensions. Note that this can be a colon-separated list of
# paths. For example: api_extensions_path =
# extensions:/path/to/more/exts:/even/more/exts. The __path__ of
# neutron.extensions is appended to this, so if your extensions are in there
# you don't need to specify them here. (string value)
#api_extensions_path =

# The type of authentication to use (string value)
auth_strategy = keystone
```

```
[keystone_authtoken]
```

```
...
```

```
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = NEUTRON_PASS
```

```
[keystone_authtoken]
```

```
#
# From keystonemiddleware.auth_token
#

# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users.
# Unauthenticated clients are redirected to this endpoint to authenticate.
# Although this endpoint should ideally be unversioned, client support in the
# wild varies. If you're using a versioned v2 endpoint here, then this should
# *not* be the same endpoint the service user utilizes for validating tokens,
# because normal end users may not be able to reach that endpoint. (string
# value)
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = 123456
```

```
NEUTRON_PASS: 123456
```

[DEFAULT]

...

notify_nova_on_port_status_changes = True

notify_nova_on_port_data_changes = True

```
# Hostname to be used by the Neutron server, agents and services running on
# this machine. All the agents and services running on this machine must use
# the same host value. (string value)
#host = example.domain

# Send notification to nova when port status changes (boolean value)
notify_nova_on_port_status_changes = true

# Send notification to nova when port data (fixed_ips/floatingip) changes so
# nova can update its cache. (boolean value)
notify_nova_on_port_data_changes = true

# Number of seconds between sending events to nova if there are any events to
# send. (integer value)
#send_events_interval = 2
```

[nova]

...

auth_url = http://controller:35357

auth_type = password

project_domain_name = Default

user_domain_name = Default

region_name = RegionOne

project_name = service

username = nova

password=NOVA_PASS

```
# Name of nova region to use. Useful if keystone manages more than one region.
# (string value)
#region_name = <None>

# Type of the nova endpoint to use. This endpoint will be looked up in the
# keystone catalog and should be one of public, internal or admin. (string
# value)
# Allowed values: public, admin, internal
#endpoint_type = public
#
# From nova.auth
#
# Authentication URL (string value)
#auth_url = <None>

auth_url = http://controller:35357
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = nova
password = 123456
```

NOVA_PASS: 123456

Διαμόρφωση Modular Layer 2 (ML2) plug-in

Παραμετροποίηση του αρχείου /etc/neutron/plugins/ml2/ml2_conf.ini

➤ root@Controller:~# vim /etc/neutron/plugins/ml2/ml2_conf.ini

```
[ml2]
```

```
...
```

```
type_drivers = flat,vlan,vxlan
```

```
[ml2]
#
# From neutron.ml2
#
# List of network type driver entrypoints to be loaded from the
# neutron.ml2.type_drivers namespace. (list value)
type_drivers = flat,vlan,vxlan
```

```
[ml2]
```

```
...
```

```
tenant_network_types = vxlan
```

```
# Ordered list of network_types to allocate as tenant networks. The default
# value 'local' is useful for single-box testing but provides no connectivity
# between hosts. (list value)
tenant_network_types = vxlan
```

```
[ml2]
```

```
...
```

```
mechanism_drivers = linuxbridge,l2population
```

```
# An ordered list of networking mechanism driver entrypoints to be loaded from
# the neutron.ml2.mechanism_drivers namespace. (list value)
mechanism_drivers = linuxbridge,l2population
```

```
[ml2]
```

```
...
```

```
extension_drivers = port_security
```

```
# An ordered list of extension driver entrypoints to be loaded from the
# neutron.ml2.extension_drivers namespace. For example: extension_drivers =
# port_security,qos (list value)
extension_drivers = port_security
```

```
[ml2_type_flat]
```

```
...
```

```
flat_networks = provider
```

```
[ml2_type_flat]
#
# From neutron.ml2
#
# List of physical_network names with which flat networks can be created. Use
# default '*' to allow flat networks with arbitrary physical_network names. Use
# an empty list to disable flat networks. (list value)
flat_networks = provider
```

[ml2_type_vxlan]

...

vni_ranges = 1:1000

```
[ml2_type_vxlan]
#
# From neutron.ml2
#
# Comma-separated list of <vni_min>:<vni_max> tuples enumerating ranges of
# VXLAN VNI IDs that are available for tenant network allocation (list value)
vni_ranges = 1:1000
# Multicast group for VXLAN. When configured, will enable sending all broadcast
# traffic to this multicast group. When left unconfigured, will disable
# multicast VXLAN mode. (string value)
#vxlan_group = <None>
```

[securitygroup]

...

enable_ipset = True

```
[securitygroup]
#
# From neutron.ml2
#
# Driver for security groups firewall in the L2 agent (string value)
#firewall_driver = <None>
# Controls whether the neutron security group API is enabled in the server. It
# should be false when using no security groups or using the nova security
# group API. (boolean value)
#enable_security_group = true
# Use ipset to speed-up the iptables based security groups. Enabling ipset
# support requires that ipset is installed on L2 agent node. (boolean value)
enable_ipset = true
```

Διαμόρφωση Linux bridge agent

Παραμετροποίηση του αρχείου /etc/neutron/plugins/ml2/linuxbridge_agent.ini

➤ root@Controller:~# vim /etc/neutron/plugins/ml2/linuxbridge_agent.ini

```
[linux_bridge]
```

...

physical_interface_mappings = provider:PROVIDER_INTERFACE_NAME

```
[linux_bridge]
#
# From neutron.ml2.linuxbridge.agent
#
# Comma-separated list of <physical_network>:<physical_interface> tuples
# mapping physical network names to the agent's node-specific physical network
# interfaces to be used for flat and VLAN networks. All physical networks
# listed in network_vlan_ranges on the server should have mappings to
# appropriate interfaces on each agent. (list value)
physical_interface_mappings = provider:enp5s0f1
```

PROVIDER_INTERFACE_NAME: enp5s0f1

[vxlan]

...

enable_vxlan = True

local_ip = OVERLAY_INTERFACE_IP_ADDRESS

l2_population = True

```
[vxlan]
#
# From neutron.ml2.linuxbridge.agent
#
# Enable VXLAN on the agent. Can be enabled when agent is managed by ml2 plugin
# using linuxbridge mechanism driver (boolean value)
enable_vxlan = true

# TTL for vxlan interface protocol packets. (integer value)
#ttl = <None>

# TOS for vxlan interface protocol packets. (integer value)
#tos = <None>

# Multicast group(s) for vxlan interface. A range of group addresses may be
# specified by using CIDR notation. Specifying a range allows different VNIs to
# use different group addresses, reducing or eliminating spurious broadcast
# traffic to the tunnel endpoints. To reserve a unique group for each possible
# (24-bit) VNI, use a /8 such as 239.0.0.0/8. This setting must be the same on
# all the agents. (string value)
#vxlan_group = 224.0.0.1

# IP address of local overlay (tunnel) network endpoint. Use either an IPv4 or
# IPv6 address that resides on one of the host network interfaces. The IP
# version of this value must match the value of the 'overlay_ip_version' option
# in the ML2 plug-in configuration file on the neutron server node(s). (IP
# address value)
local_ip = 172.16.10.231

# Extension to use alongside ml2 plugin's l2population mechanism driver. It
# enables the plugin to populate VXLAN forwarding table. (boolean value)
l2_population = true
```

OVERLAY_INTERFACE_IP_ADDRESS: 172.16.10.231

[securitygroup]

...

enable_security_group = True

firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver


```
[securitygroup]
#
# From neutron.ml2.linuxbridge.agent
#
# Driver for security groups firewall in the L2 agent (string value)
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
# Controls whether the neutron security group API is enabled in the server. It
# should be false when using no security groups or using the nova security
# group API. (boolean value)
enable_security_group = true
# Use ipset to speed-up the iptables based security groups. Enabling ipset
# support requires that ipset is installed on L2 agent node. (boolean value)
enable_ipset = true
```

Διαμόρφωση layer-3 agent

Παραμετροποίηση του αρχείου /etc/neutron/l3_agent.ini

➤ root@Controller:~# vim /etc/neutron/l3_agent.ini

```
[DEFAULT]
```

```
...
```

```
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver
```

```
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
```

```
enable_isolated_metadata = True
```

```
[DEFAULT]

#
# From neutron.base.agent
#

# Name of Open vSwitch bridge to use (string value)
#ovs_integration_bridge = br-int

# Uses veth for an OVS interface or not. Support kernels with limited namespace
# support (e.g. RHEL 6.5) so long as ovs_use_veth is set to True. (boolean
# value)
#ovs_use_veth = false

# The driver used to manage the virtual interface. (string value)
interface_driver = neutron.agent.linux.interface.BridgeInterfaceDriver

# Timeout in seconds for ovs-vsctl commands. If the timeout expires, ovs
# commands will fail with ALARMCLOCK error. (integer value)
#ovs_vsctl_timeout = 10

#
# From neutron.dhcp.agent
#

# The DHCP agent will resync its state with Neutron to recover from any
# transient notification or RPC errors. The interval is number of seconds
# between attempts. (integer value)
#resync_interval = 5

# The driver used to manage the DHCP server. (string value)
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq

# The DHCP server can assist with providing metadata support on isolated
# networks. Setting this value to True will cause the DHCP server to append
# specific host routes to the DHCP request. The metadata service will only be
# activated when the subnet does not contain any router port. The guest
# instance must be configured to request host routes via DHCP (Option 121).
# This option doesn't have any effect when force_metadata is set to True.
# (boolean value)
enable_isolated_metadata = true

# In some cases the Neutron router is not present to provide the metadata IP
# but the DHCP server can be used to provide this info. Setting this value will
# force the DHCP server to append specific host routes to the DHCP request. If
# this option is set, then the metadata service will be activated for all the
# networks. (boolean value)
#force_metadata = false

# Allows for serving metadata requests coming from a dedicated metadata access
# network whose CIDR is 169.254.169.254/16 (or larger prefix), and is connected
# to a Neutron router from which the VMs send metadata:1 request. In this case
# DHCP Option 121 will not be injected in VMs, as they will be able to reach
# 169.254.169.254 through a router. This option requires
# enable_isolated_metadata = True. (boolean value)
#enable_metadata_network = false
```

Διαμόρφωση metadata agent

Παραμετροποίηση του αρχείου /etc/neutron/metadata_agent.ini

➤ root@Controller:~# vim /etc/neutron/metadata_agent.ini

```
[DEFAULT]
...
nova_metadata_ip = controller
metadata_proxy_shared_secret = METADATA_SECRET
```

```
[DEFAULT]

#
# From neutron.metadata.agent
#

# Location for Metadata Proxy UNIX domain socket. (string value)
#metadata_proxy_socket = $state_path/metadata_proxy

# User (uid or name) running metadata proxy after its initialization (if empty:
# agent effective user). (string value)
#metadata_proxy_user =

# Group (gid or name) running metadata proxy after its initialization (if
# empty: agent effective group). (string value)
#metadata_proxy_group =

# Certificate Authority public key (CA cert) file for ssl (string value)
#auth_ca_cert = <None>

# IP address used by Nova metadata server. (string value)
nova_metadata_ip = controller

# TCP Port used by Nova metadata server. (port value)
# Minimum value: 0
# Maximum value: 65535
#nova_metadata_port = 8775

# When proxying metadata requests, Neutron signs the Instance-ID header with a
# shared secret to prevent spoofing. You may select any string for a secret,
# but it must match here and in the configuration used by the Nova Metadata
# Server. NOTE: Nova uses the same config key, but in [neutron] section.
# (string value)
metadata_proxy_shared_secret = 123456
```

METADATA_SECRET: 123456

Διαμόρφωση Compute service για την χρήση του Networking service

Παραμετροποίηση του αρχείου /etc/nova/nova.conf

➤ root@Controller:~# vim /etc/nova/nova.conf

```
[neutron]
...
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = NEUTRON_PASS
service_metadata_proxy = True
metadata_proxy_shared_secret = METADATA_SECRET
```

```
[neutron]
#
# Configuration options for neutron (network connectivity as a service).
#
# From nova.conf
#
#
# This option specifies the URL for connecting to Neutron.
#
# Possible values:
#
# * Any valid URL that points to the Neutron API service is appropriate here.
# This typically matches the URL returned for the 'network' service type
# from the Keystone service catalog.
# (uri value)
url=http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = 123456
service_metadata_proxy = True
metadata_proxy_shared_secret = 123456
```

NEUTRON_PASS: 123456

METADATA_SECRET: 123456

4.11.3 Ολοκλήρωση εγκατάστασης

Populate the database

- root@Controller:~# su -s /bin/sh -c "neutron-db-manage --config-file \ /etc/neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/ml2_conf.ini \ upgrade head" neutron

```
OK
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
Running upgrade for neutron-fwaas ...
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
INFO [alembic.runtime.migration] Running upgrade -> start_neutron_fwaas, start neutron-fwaas chain
INFO [alembic.runtime.migration] Running upgrade start_neutron_fwaas -> 4202e3047e47, add_index tenant_id
INFO [alembic.runtime.migration] Running upgrade 4202e3047e47 -> 540142f314f4, FWaaS router insertion
INFO [alembic.runtime.migration] Running upgrade 540142f314f4 -> 796c68dffbb, cisco_csr_fwaas
INFO [alembic.runtime.migration] Running upgrade 796c68dffbb -> kilo, kilo
INFO [alembic.runtime.migration] Running upgrade kilo -> 67c8e8d61d5, Initial Liberty no-op script.
INFO [alembic.runtime.migration] Running upgrade 67c8e8d61d5 -> 458aa42b14b, fw_table_alter script to make <name> column case sensitive
INFO [alembic.runtime.migration] Running upgrade 458aa42b14b -> f83a0b2964d0, rename tenant to project
INFO [alembic.runtime.migration] Running upgrade kilo -> c40fbb377ad, Initial Liberty no-op script.
INFO [alembic.runtime.migration] Running upgrade c40fbb377ad -> 4b47ea298795, add reject rule
INFO [alembic.runtime.migration] Running upgrade 4b47ea298795 -> d6a12e637e28, neutron-fwaas v2.0
OK
root@Controller:~#
```

Επανεκκίνηση Compute API service & Networking services

- root@Controller:~# service nova-api restart
- root@Controller:~# service neutron-server restart
- root@Controller:~# service neutron-linuxbridge-agent restart
- root@Controller:~# service neutron-dhcp-agent restart
- root@Controller:~# service neutron-metadata-agent restart
- root@Controller:~# service neutron-l3-agent restart

4.11.4 Εγκατάσταση και διαμόρφωση – Compute node

Εγκατάσταση πακέτων

➤ root@Compute:~# apt install neutron-linuxbridge-agent

Διαμορφώση

Παραμετροποίηση του αρχείου /etc/neutron/neutron.conf

➤ root@Compute:~# vim /etc/neutron/neutron.conf

Στο section [Database] αφαιρούμε οποιαδήποτε σύνδεση υπάρχει

```
# The SQLAlchemy connection string to use to connect to the database. (string
# value)
# Deprecated group/name - [DEFAULT]/sql_connection
# Deprecated group/name - [DATABASE]/sql_connection
# Deprecated group/name - [sql]/connection
connection = sqlite:///var/lib/neutron/neutron.sqlite
```

[DEFAULT]

...

transport_url = rabbit://openstack:RABBIT_PASS@controller

```
# A URL representing the messaging driver to use and its full configuration.
# (string value)
transport_url = rabbit://openstack:123456@controller

# DEPRECATED: The messaging driver to use defaults to rabbit. Other drivers
# include amqp and zmq. (string value)
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Replaced by [DEFAULT]/transport_url
#rpc_backend = rabbit
```

RABBIT_PASS:123456

[DEFAULT]

...

auth_strategy = keystone

```
# The port to bind to (port value)
# Minimum value: 0
# Maximum value: 65535
#bind_port = 9696

# The path for API extensions. Note that this can be a colon-separated list of
# paths. For example: api_extensions_path =
# extensions:/path/to/more/exts:/even/more/exts. The __path__ of
# neutron.extensions is appended to this, so if your extensions are in there
# you don't need to specify them here. (string value)
#api_extensions_path =

# The type of authentication to use (string value)
auth_strategy = keystone
```

```
[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = NEUTRON_PASS
```

```
[keystone_authtoken]
#
# From keystonemiddleware.auth_token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users.
# Unauthenticated clients are redirected to this endpoint to authenticate.
# Although this endpoint should ideally be unversioned, client support in the
# wild varies. If you're using a versioned v2 endpoint here, then this should
# *not* be the same endpoint the service user utilizes for validating tokens,
# because normal end users may not be able to reach that endpoint. (string
# value)
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = neutron
password = 123456
```

NEUTRON_PASS: 123456

4.11.5 Διαμόρφωση δικτύου

Διαμόρφωση Linux bridge agent

Παραμετροποίηση του αρχείου /etc/neutron/plugins/ml2/linuxbridge_agent.ini

➤ root@Compute:~# vim /etc/neutron/plugins/ml2/linuxbridge_agent.ini

```
[linux_bridge]
```

...

```
physical_interface_mappings = provider:PROVIDER_INTERFACE_NAME
```

```
[linux_bridge]
#
# From neutron.ml2.linuxbridge.agent
#
# Comma-separated list of <physical_network>:<physical_interface> tuples
# mapping physical network names to the agent's node-specific physical network
# interfaces to be used for flat and VLAN networks. All physical networks
# listed in network_vlan_ranges on the server should have mappings to
# appropriate interfaces on each agent. (list value)
physical_interface_mappings = provider:enp2s0
#
# List of <physical_network>:<physical_bridge> (list value)
#bridge_mappings =
```

PROVIDER_INTERFACE_NAME: **enp2s0**

```
[vxlan]
...
enable_vxlan = True
local_ip = OVERLAY_INTERFACE_IP_ADDRESS...
l2_population = True
```

```
[vxlan]
#
# From neutron.ml2.linuxbridge.agent
#
# Enable VXLAN on the agent. Can be enabled when agent is managed by ml2 plugin
# using linuxbridge mechanism driver (boolean value)
enable_vxlan = true
```

```
# IP address of local overlay (tunnel) network endpoint. Use either an IPv4 or
# IPv6 address that resides on one of the host network interfaces. The IP
# version of this value must match the value of the 'overlay_ip_version' option
# in the ML2 plug-in configuration file on the neutron server node(s). (IP
# address value)
local_ip = 172.16.10.232
# Extension to use alongside ml2 plugin's l2population mechanism driver. It
# enables the plugin to populate VXLAN forwarding table. (boolean value)
l2_population = true
```

OVERLAY_INTERFACE_IP_ADDRESS: **172.16.10.232**

```
[securitygroup]
...
enable_security_group = True
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```



```
[securitygroup]
#
# From neutron.ml2.linuxbridge.agent
#
# Driver for security groups firewall in the L2 agent (string value)
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
# Controls whether the neutron security group API is enabled in the server. It
# should be false when using no security groups or using the nova security
# group API. (boolean value)
enable_security_group = true
# Use ipset to speed-up the iptables based security groups. Enabling ipset
# support requires that ipset is installed on L2 agent node. (boolean value)
enable_ipset = true
```

Διαμόρφωση Compute service

Παραμετροποίηση του αρχείου /etc/nova/nova.conf

➤ root@Compute:~# vim /etc/nova/nova.conf

```
[neutron]
...
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = NEUTRON_PASS
```

```
[neutron]
#
# Configuration options for neutron (network connectivity as a service).
#
# From nova.conf
#
#
# This option specifies the URL for connecting to Neutron.
#
# Possible values:
#
# * Any valid URL that points to the Neutron API service is appropriate here.
#   This typically matches the URL returned for the 'network' service type
#   from the Keystone service catalog.
# (uri value)
url=http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = Default
user_domain_name = Default
region_name = RegionOne
project_name = service
username = neutron
password = 123456
```

NEUTRON_PASS: 123456

4.11.6 Ολοκλήρωση εγκατάστασης

- ✓ Επανεκκίνηση nova-compute και neutron-linuxbridge-agent services
- root@Compute:~# service nova-compute restart
- root@Compute:~# service neutron-linuxbridge-agent restart

Επαλήθευση λειτουργίας

Admin credentials

- root@Controller:~# . admin-openrc.sh

Neutron-server process

- root@Controller:~# neutron ext-list

```

root@Controller:~# . admin-openrc.sh
root@Controller:~# neutron ext-list
neutron CLI is deprecated and will be removed in the future. Use openstack CLI instead.
+-----+-----+
| alias                | name                |
+-----+-----+
| default-subnetpools  | Default Subnetpools|
| network-ip-availability | Network IP Availability|
| network_availability_zone | Network Availability Zone|
| auto-allocated-topology | Auto Allocated Topology Services|
| ext-gw-mode          | Neutron L3 Configurable external gateway mode|
| binding              | Port Binding        |
| agent                | agent               |
| subnet_allocation    | Subnet Allocation   |
| l3_agent_scheduler   | L3 Agent Scheduler  |
| tag                  | Tag support         |
| external-net         | Neutron external network|
| flavors              | Neutron Service Flavors|
| net-mtu              | Network MTU        |
| availability_zone    | Availability Zone   |
| quotas               | Quota management support|
| l3-ha                 | HA Router extension|
| provider             | Provider Network   |
| multi-provider       | Multi Provider Network|
| address-scope        | Address scope       |
| extraroute           | Neutron Extra Route|
| subnet-service-types | Subnet service types|
| standard-attr-timestamp | Resource timestamps|
| service-type         | Neutron Service Type Management|
| l3-flavors           | Router Flavor Extension|
| port-security        | Port Security       |
| extra_dhcp_opt       | Neutron Extra DHCP opts|
| standard-attr-revisions | Resource revision numbers|
| pagination           | Pagination support  |
| sorting              | Sorting support     |
| security-group       | security-group      |
| dhcp_agent_scheduler | DHCP Agent Scheduler|
| router_availability_zone | Router Availability Zone|
| rbac-policies        | RBAC Policies      |
| tag-ext              | Tag support for resources: subnet, subnetpool, port, router|
| standard-attr-description | standard-attr-description|
| router               | Neutron L3 Router   |
| allowed-address-pairs | Allowed Address Pairs|
| project-id           | project_id field enabled|
| dvr                  | Distributed Virtual Router|
+-----+-----+
root@Controller:~# █

```

- root@Controller:~# openstack network agent list

```
root@Controller:~# openstack network agent list
```

ID	Agent Type	Host	Availability Zone	Alive	State	Binary
517b38d4-f49d-4e2d-a7b0-72b5da0f9176	Linux bridge agent	Controller	None	True	UP	neutron-linuxbridge-agent
7dbe817a-299f-4aa0-b03f-c1d43d9ad61d	Metadata agent	Controller	None	True	UP	neutron-metadata-agent
850e7b0f-10ab-4374-9f40-f1626e4be198	DHCP agent	controller	nova	True	UP	neutron-dhcp-agent
912e741f-1e89-45ef-a81b-841d6d21c89d	Linux bridge agent	Compute	None	True	UP	neutron-linuxbridge-agent
b5cacc94-1d1b-4c66-be77-b6ea8e57abc3	L3 agent	Controller	nova	True	UP	neutron-l3-agent

```
root@Controller:~#
```

4.12 Dashboard

Το Dashboard (horizon) είναι ένα web interface που επιτρέπει στους διαχειριστές του cloud και στους χρήστες να διαχειρίζονται διάφορους πόρους και υπηρεσίες του OpenStack.

Αυτή η ενότητα περιγράφει τον τρόπο εγκατάστασης και ρύθμισης παραμέτρων του πίνακα ελέγχου - dashboard στον Controller node.

Η μόνη βασική υπηρεσία που απαιτείται από τον πίνακα ελέγχου είναι η υπηρεσία Ταυτότητας - Identity service. (guide, n.d.)

4.12.1 Εγκατάσταση και διαμόρφωση

Εγκατάσταση πακέτων

➤ root@Controller:~# apt install openstack-dashboard

Παραμετροποίηση του αρχείου /etc/openstack-dashboard/local_settings.py

➤ root@Controller:~# vim /etc/openstack-dashboard/local_settings.py

OPENSTACK_HOST = "controller"

```
# For multiple regions uncomment this configuration, and add (endpoint, title).
#AVAILABLE_REGIONS = [
#    ('http://cluster1.example.com:5000/v2.0', 'cluster1'),
#    ('http://cluster2.example.com:5000/v2.0', 'cluster2'),
#]

OPENSTACK_HOST = "controller"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v2.0" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "_member_"
```

ALLOWED_HOSTS = ['*',]

```
# If horizon is running in production (DEBUG is False), set this
# with the list of host/domain names that the application can serve.
# For more information see:
# https://docs.djangoproject.com/en/dev/ref/settings/#allowed-hosts
ALLOWED_HOSTS = ['*', ]
```

SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

```
CACHES = {
    'default': {
```

```
'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
'LOCATION': 'controller:11211',
}
}
```

```
# We recommend you use memcached for development; otherwise after every reload
# of the django development server, you will have to login again. To use
# memcached set CACHES to something like

SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': 'controller:11211',
    }
}

#CACHES = {
#    'default': {
#        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
#        'LOCATION': '127.0.0.1:11211',
#    },
#}
```

Identity API version 3

```
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
```

```
OPENSTACK_HOST = "controller"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "_member_"

# Enables keystone web single-sign-on if set to True.
#WEBSO_ENABLED = False
```

```
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
```

```
# Set this to True if running on a multi-domain model. When this is enabled, it
# will require the user to enter the Domain name in addition to the username
# for login.
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
```

```
OPENSTACK_API_VERSIONS = {
    "identity": 3,
    "image": 2,
    "volume": 2,
}
```

```
# NOTE: The version should be formatted as it appears in the URL for the
# service API. For example, The identity service APIs have inconsistent
# use of the decimal point, so valid options would be 2.0 or 3.
# Minimum compute version to get the instance locked status is 2.9.
OPENSTACK_API_VERSIONS = {
#    "data-processing": 1.1,
#    "identity": 3,
#    "image": 2,
#    "volume": 2,
#    "compute": 2,
}
```

“default” as the default domain

OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "default"

```
# NOTE: This value must be the name of the default domain, NOT the ID.
# Also, you will most likely have a value in the keystone policy file like this
#   "cloud_admin": "rule:admin_required and domain_id:<your domain id>"
# This value must be the name of the domain whose ID is specified there.
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
```

user as the default role

OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

```
#
OPENSTACK_HOST = "controller"
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

# Enables keystone web single-sign-on if set to True.
#WEBSO_ENABLED = False
```

TIME_ZONE = "TIME_ZONE"

```
# The timezone of the server. This should correspond with the timezone
# of your entire OpenStack installation, and hopefully be in UTC.
TIME_ZONE = "Europe/Athens"
```

✓ Αλλαγή δικαιωμάτων από root σε www-data στο αρχείο /var/lib/openstack-dashboard/secret_key

➤ root@Controller:~#chown -R www-data:www-data /var/lib/openstack-dashboard/secret_key

➤ root@Controller:~# chmod 600 /var/lib/openstack-dashboard/secret_key

```
drwxr-xr-x  4 horizon  horizon  4096 Jul 25 17:04 ./
drwxr-xr-x 78 root      root      4096 Jul 22 21:57 ../
-rw-----  1 www-data www-data   64 Jul 25 17:04 secret_key
drwxr-xr-x  2 root      root      4096 Jun 19 10:55 secret-key/
drwxr-xr-x 10 www-data www-data  4096 Jul 25 17:05 static/
-rw-r--r--  1 horizon  horizon    0 Jul 22 18:35 _var_lib_openstack-dashboard_secret_key.lock
```

Παραμετροποίηση του αρχείου /etc/apache2/conf-available/openstack-dashboard.conf προσθέτοντας την παρακάτω γραμμή

WSGIApplicationGroup %{GLOBAL}

```
WSGIScriptAlias /horizon /usr/share/openstack-dashboard/openstack_dashboard/wsgi/django.wsgi process-group=horizon
WSGIDaemonProcess horizon user=www-data group=www-data processes=3 threads=10 display-name=%{GROUP}
WSGIProcessGroup horizon
WSGIApplicationGroup %{GLOBAL}

Alias /static /var/lib/openstack-dashboard/static/
Alias /horizon/static /var/lib/openstack-dashboard/static/

<Directory /usr/share/openstack-dashboard/openstack_dashboard/wsgi>
    Require all granted
</Directory>

<Directory /var/lib/openstack-dashboard/static>
    Require all granted
</Directory>
```

➤ root@Controller:~# vim /etc/apache2/conf-available/openstack-dashboard.conf

4.12.2 Ολοκλήρωση εγκατάστασης

➤ root@Controller:~# service apache2 reload

Επαλήθευση λειτουργίας

Για την επαλήθευση της λειτουργίας κάνουμε launch από browser το παρακάτω link με τους χρήστες που έχουμε δημιουργήσει

<http://controller/horizon> ή <http://172.16.10.231/horizon>

Domain:	default	default	default
Username:	admin	demo	kmarinis
Password:	123456	123456	123456

Login Page

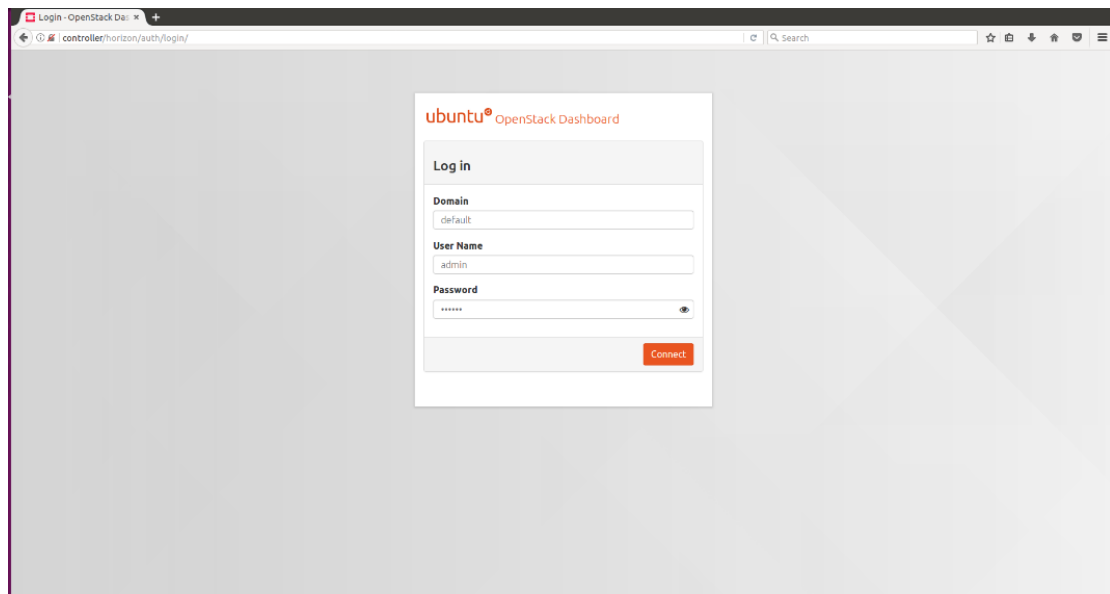


Figure 12: OpenStack Dashboard

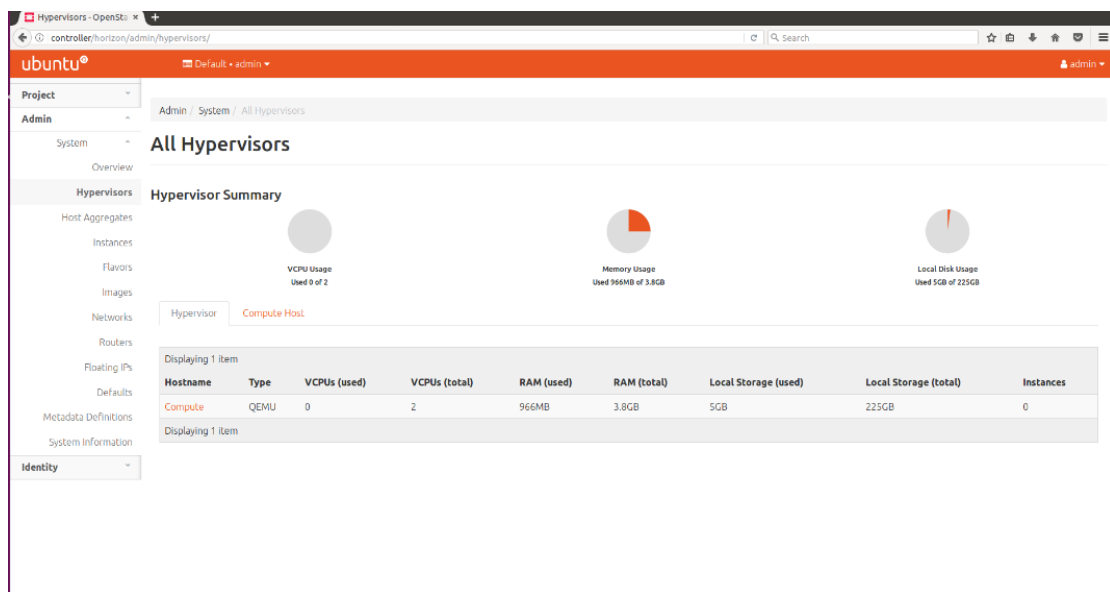


Figure 13: OpenStack Dashboard

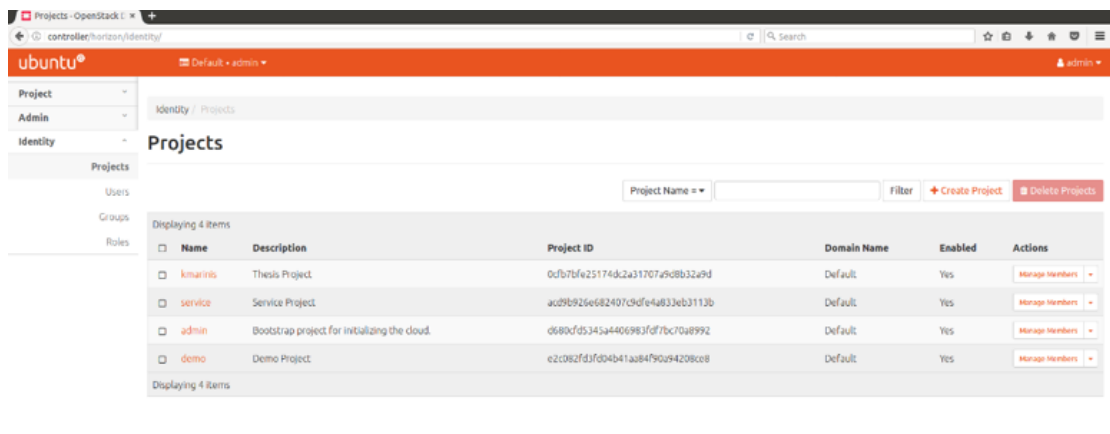


Figure 14: OpenStack Dashboard

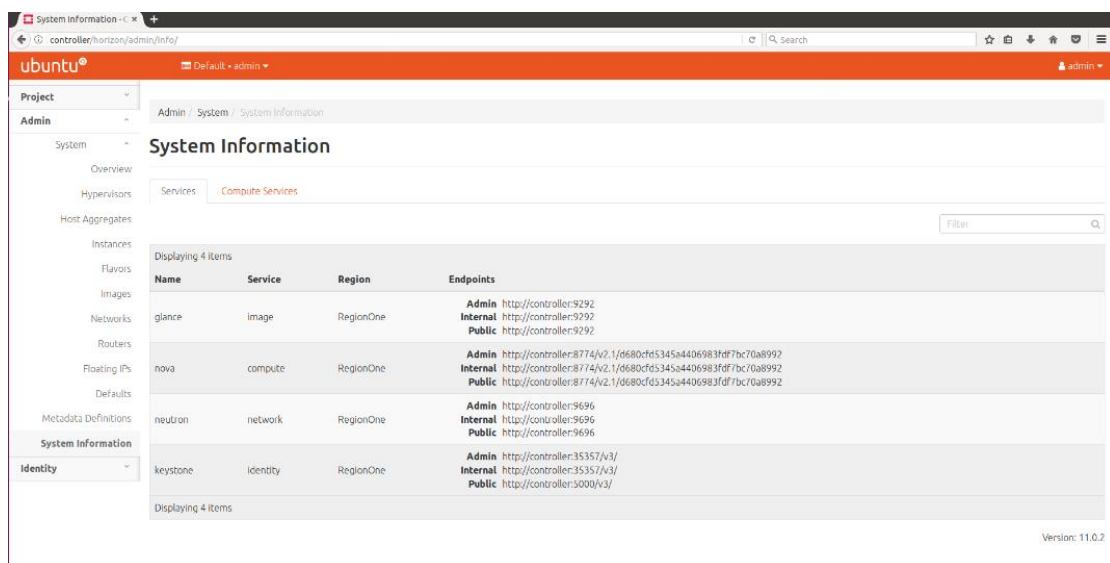


Figure 15: OpenStack Dashboard

4.13 Block Storage service

Η υπηρεσία Block Storage (cinder) παρέχει συσκευές αποθήκευσης μπλοκ σε επισκέψεις επισκέψεων. Η μέθοδος κατά την οποία η αποθήκευση παρέχεται και καταναλώνεται καθορίζεται από το πρόγραμμα οδήγησης Block Storage ή τα προγράμματα οδήγησης σε περίπτωση διαμόρφωσης πολλαπλών αντιγράφων. Υπάρχουν διάφορα προγράμματα οδήγησης που είναι διαθέσιμα: NAS / SAN, NFS, iSCSI, Ceph και πολλά άλλα.

Το API Block Storage API και οι υπηρεσίες προγραμματισμού συνήθως εκτελούνται στους κόμβους του ελεγκτή. Ανάλογα με τα προγράμματα οδήγησης που χρησιμοποιούνται, η υπηρεσία όγκου μπορεί να εκτελεστεί σε controller nodes, compute nodes και standalone storage nodes.

Η υπηρεσία OpenStack Block Storage (cinder) προσθέτει συνεχόμενο αποθηκευτικό χώρο σε ένα εικονικό μηχάνημα. Η Αποθήκευση Αποθήκευσης παρέχει μια υποδομή για τη διαχείριση τόμων και αλληλεπιδρά με το OpenStack Compute για την παροχή τόμων για περιπτώσεις. Η υπηρεσία επιτρέπει επίσης τη διαχείριση στιγμιότυπων όγκου και τύπων όγκου. (guide, n.d.)

Η υπηρεσία Block Storage αποτελείται από τα ακόλουθα στοιχεία

cinder-api

Αποδέχεται τα αιτήματα API και τις μεταφέρει στον όγκο σκωρίας για δράση.

cinder-volume

Απευθύνεται άμεσα στην υπηρεσία Block Storage και διεργασίες όπως ο cinder-scheduler. Επίσης, αλληλεπιδρά με αυτές τις διαδικασίες μέσω μιας ουράς μηνυμάτων. Η υπηρεσία cinder-volume ανταποκρίνεται στις αιτήσεις ανάγνωσης και εγγραφής που αποστέλλονται στην υπηρεσία αποθήκευσης μπλοκ για τη διατήρηση της κατάστασης. Μπορεί να αλληλεπιδράσει με διάφορους παρόχους αποθήκευσης μέσω αρχιτεκτονικής προγράμματος οδήγησης.

cinder-scheduler daemon

Επιλέγει τον βέλτιστο κόμβο προμηθευτή αποθήκευσης στον οποίο θα δημιουργηθεί η ένταση. Μια παρόμοια συνιστώσα με το nova-scheduler.

cinder-backup daemon

Η υπηρεσία δημιουργίας εφεδρικών αντιγράφων παρέχει τη δημιουργία αντιγράφων ασφαλείας οποιωνδήποτε τύπων σε έναν εφεδρικό παροχέα αποθήκευσης. Όπως και η υπηρεσία cinder-volume, μπορεί να αλληλεπιδράσει με διάφορους παρόχους αποθήκευσης μέσω αρχιτεκτονικής προγράμματος οδήγησης.

Messaging queue

Διαδρομές πληροφορίες μεταξύ των διαδικασιών αποθήκευσης Block. (guide, n.d.)

4.13.1 Εγκατάσταση και διαμόρφωση

Δημιουργία βάσης δεδομένων - **cinder**

➤ root@Controller:~# mysql -u root -p

```
MariaDB [(none)]> CREATE DATABASE cinder;
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' \
IDENTIFIED BY 'CINDER_DBPASS';
```

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' \
IDENTIFIED BY CINDER_DBPASS';
```

```
MariaDB [(none)]> exit
```

```
root@Controller:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 64
Server version: 10.0.29-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE cinder;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' \
-> IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' \
-> IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
root@Controller:~#
```

CINDER_DBPASS: **123456**

Admin credentials

➤ root@Controller:~# . admin-openrc

Service credentials – Δημιουργία

cinder user

➤ root@Controller:~# openstack user create --domain default \
--password-prompt cinder

```

root@Controller:~# . admin-openrc
root@Controller:~# openstack user create --domain default --password-prompt cinder
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | default |
| enabled | True |
| id | 439e7fdbf3a649bebd24e4b |
| name | cinder |
| options | {} |
| password_expires_at | None |
+-----+-----+

```

Πρόσθεση ρόλου admin στον cinder user στο service project

➤ root@Controller:~# openstack role add --project service --user cinder admin

Δημιουργία cinder και cinderv2 service

➤ root@Controller:~# openstack service create --name cinder \
 --description "OpenStack Block Storage" volume

```

root@Controller:~# openstack role add --project service --user cinder admin
root@Controller:~# openstack service create --name cinder \  

> --description "OpenStack Block Storage" volume
+-----+-----+
| Field | Value |
+-----+-----+
| description | OpenStack Block Storage |
| enabled | True |
| id | e93ec21d2bfb47e88ab2b3add89846f2 |
| name | cinder |
| type | volume |
+-----+-----+

```

➤ root@Controller:~# openstack service create --name cinderv2 \
 --description "OpenStack Block Storage" volumev2

```

root@Controller:~# openstack service create --name cinderv2 \  

> --description "OpenStack Block Storage" volumev2
+-----+-----+
| Field | Value |
+-----+-----+
| description | OpenStack Block Storage |
| enabled | True |
| id | fad18c2a65484bfc13b623d6768a6fc |
| name | cinderv2 |
| type | volumev2 |
+-----+-----+

```

✓ To Block Storage services απαιτεί δημιουργία δύο service entities.

Δημιουργία Block Storage service API endpoints - **cinder**

➤ root@Controller:~# openstack endpoint create --region RegionOne \
 volume public http://controller:8776/v1/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne \
> volume public http://controller:8776/v1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 62e47ed15ba3474ebed0ec64024a3b7e |
| interface | public |
| region | RegionOne |
| region_id | RegionOne |
| service_id | e93ec21d2bfb47e88ab2b3add89846f2 |
| service_name | cinder |
| service_type | volume |
| url | http://controller:8776/v1/%(tenant_id)s |
+-----+-----+

```

- root@Controller:~# openstack endpoint create --region RegionOne \
 volume internal http://controller:8776/v1/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne \
> volume internal http://controller:8776/v1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | e92692de382f42d1b2886de48a07ea57 |
| interface | internal |
| region | RegionOne |
| region_id | RegionOne |
| service_id | e93ec21d2bfb47e88ab2b3add89846f2 |
| service_name | cinder |
| service_type | volume |
| url | http://controller:8776/v1/%(tenant_id)s |
+-----+-----+

```

- root@Controller:~# openstack endpoint create --region RegionOne \
 volume admin http://controller:8776/v1/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne \
> volume admin http://controller:8776/v1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 1a15f39f9548400bb2001d0e06b01430 |
| interface | admin |
| region | RegionOne |
| region_id | RegionOne |
| service_id | e93ec21d2bfb47e88ab2b3add89846f2 |
| service_name | cinder |
| service_type | volume |
| url | http://controller:8776/v1/%(tenant_id)s |
+-----+-----+

```

Δημιουργία Block Storage service API endpoints – **cinderv2**

- root@Controller:~# openstack endpoint create --region RegionOne \
 volumev2 public http://controller:8776/v2/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne \
> volumev2 public http://controller:8776/v2/%(tenant_id)s
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | 94e080d5824c4bfa8dfd83d558387505       |
| interface  | public                                   |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | fad18c2a65484bfc13b623d6768a6fc       |
| service_name | cinderv2                                |
| service_type | volumev2                                 |
| url        | http://controller:8776/v2/%(tenant_id)s |
+-----+-----+
  
```

- root@Controller:~# openstack endpoint create --region RegionOne \
 volumev2 internal http://controller:8776/v2/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne \
> volumev2 internal http://controller:8776/v2/%(tenant_id)s
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | 6130149516e34724a5ba1a37cfdeaa10     |
| interface  | internal                                 |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | fad18c2a65484bfc13b623d6768a6fc       |
| service_name | cinderv2                                |
| service_type | volumev2                                 |
| url        | http://controller:8776/v2/%(tenant_id)s |
+-----+-----+
  
```

- root@Controller:~# openstack endpoint create --region RegionOne \
 volumev2 admin http://controller:8776/v2/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne \
> volumev2 admin http://controller:8776/v2/%(tenant_id)s
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | 330dc1e1b59144d0878290e9a52a9586     |
| interface  | admin                                    |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | fad18c2a65484bfc13b623d6768a6fc       |
| service_name | cinderv2                                |
| service_type | volumev2                                 |
| url        | http://controller:8776/v2/%(tenant_id)s |
+-----+-----+
  
```

4.13.2 Εγκατάσταση και διαμόρφωση Controller node

Εγκατάσταση πακέτων

```
root@Controller:~# apt install cinder-api cinder-scheduler
```

Παραμετροποίηση του αρχείου `/etc/cinder/cinder.conf`

```
➤ root@Controller:~# vim/etc/cinder/cinder.conf
```

```
[database]
```

```
...
```

```
connection = mysql+pymysql://cinder:CINDER_DBPASS@controller/cinder
```

```
[database]
connection = mysql+pymysql://cinder:123456@controller/cinder
```

Αντικατάσταση το `RABBIT_PASS` με τον κωδικό πρόσβασης που έχουμε θέσει για το λογαριασμό `openstack` στο `RabbitMQ`.

```
RABBIT_PASS: 123456
```

```
[DEFAULT]
```

```
...
```

```
my_ip = 172.16.10.231
```

```
transport_url = rabbit://openstack:RABBIT_PASS@controller
```

```
auth_strategy = keystone
```

```
[DEFAULT]
rootwrap_config = /etc/cinder/rootwrap.conf
api_paste_config = /etc/cinder/api-paste.ini
iscsi_helper = tgtadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
state_path = /var/lib/cinder
lock_path = /var/lock/cinder
volumes_dir = /var/lib/cinder/volumes

my_ip = 172.16.10.231

transport_url = rabbit://openstack:123456@controller
auth_strategy = keystone
```

```
[keystone_authtoken]
```

```
...
```

```
auth_uri = http://controller:5000
```

```
auth_url = http://controller:35357
```

```
memcached_servers = controller:11211
```

```
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
username = cinder
password = CINDER_PASS
```

Αντικατάσταση το CINDER_PASS με τον κωδικό πρόσβασης που έχουμε θέσει για τον cinder user στην υπηρεσία Ταυτότητα- Identity service.

CINDER_PASS: **123456**

```
[oslo_concurrency]
...
lock_path = /var/lib/cinder/tmp
```

```
[oslo_concurrency]
lock_path = /var/lib/cinder/tmp
```

Populate the Block Storage database

➤ root@Controller:~# su -s /bin/sh -c "cinder-manage db sync" cinder

Ρύθμιση παραμέτρων για χρήση του Block Storage από τον Compute node

Παραμετροποίηση του αρχείου /etc/nova/nova.conf

➤ root@Controller:~# vim /etc/nova/nova.conf

```
[cinder]
...
os_region_name = RegionOne
```

```
# * Any string representing region name
# (string value)
os_region_name = RegionOne
```

4.13.3 Ολοκλήρωση εγκατάστασης

Επανεκκίνηση του Compute API service

➤ root@Controller:~# service nova-api restart

Επανεκκίνηση των Block Storage services

➤ root@Controller:~# service cinder-scheduler restart

➤ root@Controller:~# service cinder-api restart

4.13.4 Εγκατάσταση και διαμόρφωση Block Storage node

Η υπηρεσία παρέχει logical volumes σε αυτήν τη συσκευή χρησιμοποιώντας τον οδηγό LVM και τους παρέχει σε περιπτώσεις μέσω μεταφοράς iSCSI. Μπορείτε να ακολουθήσετε αυτές τις οδηγίες με μικρές τροποποιήσεις για την οριζόντια κλιμάκωση του περιβάλλοντός σας με πρόσθετους κόμβους αποθήκευσης (storage nodes). (guide, n.d.)

Εγκατάσταση πακέτων

- root@BlockStorage1:~# apt install lvm2

Δημιουργία LVM physical volume /dev/sdb

- root@BlockStorage1:~# pvcreate /dev/sdb

```
root@BlockStorage1:~# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
root@BlockStorage1:~#
```

Δημιουργία LVM volume group cinder-volumes

- root@BlockStorage1:~# vgcreate cinder-volumes /dev/sdb

```
root@BlockStorage1:~# vgcreate cinder-volumes /dev/sdb
Volume group "cinder-volumes" successfully created
root@BlockStorage1:~#
```

Διαμόρφωση LVM για να βλέπει μόνο τις συσκευές που περιέχουν το cinder-volumes volume group παραμετροποιώντας το αρχείο /etc/lvm/lvm.conf

- root@BlockStorage1:~# vim /etc/lvm/lvm.conf

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

```
# Use anchors to be very specific:
filter = [ "a/sdb/", "r/.*/" ]
#
# This configuration option has an automatic default value.
# filter = [ "a|.*|" ]
```

Εγκατάσταση και διαμόρφωση components

- root@BlockStorage1:~# apt install cinder-volume

Παραμετροποίηση του αρχείου /etc/cinder/cinder.conf

- root@BlockStorage1:~# vim /etc/cinder/cinder.conf

```
[database]
```

```
...
```

```
connection = mysql+pymysql://cinder:CINDER_DBPASS@controller/cinder
```

```
[database]
connection = mysql+pymysql://cinder:123456@controller/cinder
```

CINDER_DBPASS: **123456**

```
[DEFAULT]
```

```
...
```

```
my_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
transport_url = rabbit://openstack:RABBIT_PASS@controller
auth_strategy = keystone
enabled_backends = lvm
glance_api_servers = http://controller:9292
```

```
[DEFAULT]
rootwrap_config = /etc/cinder/rootwrap.conf
api_paste_config = /etc/cinder/api-paste.ini
iscsi_helper = tgtadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
state_path = /var/lib/cinder
lock_path = /var/lock/cinder
volumes_dir = /var/lib/cinder/volumes

my_ip = 172.16.10.233

transport_url = rabbit://openstack:123456@controller

auth_strategy = keystone

enabled_backends = lvm

glance_api_servers = http://controller:9292
```

MANAGEMENT_INTERFACE_IP_ADDRESS – BlockStorage1 - my_ip: **172.16.10.233**

```
[keystone_authtoken]
```

```
...
```

```
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = Default
user_domain_name = Default
project_name = service
```

```
username = cinder  
password = CINDER_PASS
```

```
[keystone_authtoken]  
auth_uri = http://controller:5000  
auth_url = http://controller:35357  
memcached_servers = controller:11211  
auth_type = password  
project_domain_name = Default  
user_domain_name = Default  
project_name = service  
username = cinder  
password = 123456
```

CINDER_PASS: **123456**

```
[lvm]  
...  
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver  
volume_group = cinder-volumes  
iscsi_protocol = iscsi  
iscsi_helper = tgtadm
```

```
[lvm]  
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver  
volume_group = cinder-volumes  
iscsi_protocol = iscsi  
iscsi_helper = tgtadm
```

```
[oslo_concurrency]  
...  
lock_path = /var/lib/cinder/tmp
```

```
[oslo_concurrency]  
lock_path = /var/lib/cinder/tmp
```

4.13.5 Ολοκλήρωση εγκατάστασης

- root@BlockStorage1:~# service tgt restart
- root@BlockStorage1:~# service cinder-volume restart

4.13.6 Επαλήθευση λειτουργίας

Admin credentials

- root@Controller:~# . admin-openrc
- root@Controller:~# openstack volume service list

```

root@Controller:~# . admin-openrc
root@Controller:~# openstack volume service list
+-----+-----+-----+-----+-----+-----+
| Binary          | Host           | Zone | Status | State | Updated At |
+-----+-----+-----+-----+-----+-----+
| cinder-scheduler | Controller     | nova | enabled | up    | 2017-08-16T17:56:15.000000 |
| cinder-volume    | BlockStorage1@lvm | nova | enabled | up    | 2017-08-16T17:56:19.000000 |
+-----+-----+-----+-----+-----+-----+

```

4.14 Launch an instance

Δημιουργία virtual networks

- ✓ Πρέπει να δημιουργηθεί πρώτα το provider network και μετά το self-service network

Δημιουργία provider network

Admin credentials

- root@Controller:~# . admin-openrc

Δημιουργία network

- root@Controller:~# openstack net-create public --router:external --shared

Δημιουργία subnet στο network

- root@Controller:~# openstack subnet-create public 172.16.10.0/24 --name public-subnet --allocation-pool start=172.16.10.100,end=172.16.10.125 --disable-dhcp --gateway 172.16.10.1 --dns-nameserver 8.8.8.8
- root@Controller:~# openstack subnet list

```

root@Controller:~# openstack subnet list
+-----+-----+-----+-----+
| ID              | Name           | Network | Subnet |
+-----+-----+-----+-----+
| 15ea38fd-296d-4887-826d-ac4bc873370f | private-subnet1 | 71e00a67-7d2e-4c08-86e9-ac489c589ef3 | 10.0.0.0/24 |
| b550fdb6-39fd-445b-9db7-21559a874d16 | demo-subnet    | b5b025fe-5b32-4023-9421-fb1ec9bd02ea | 20.0.0.0/24 |
| d8c0c61a-1f8d-4485-803b-4ae06cba4208 | public-subnet  | 171edd41-df8c-402d-a7ea-dbf22c3dddb | 172.16.10.0/24 |
| f9c4cfff-014b-4c92-94f6-17bad8f4c425 | demo_subnet    | f3cdda4a-bc65-44fc-90f4-1ab31e0cffff | 15.0.0.0/24 |
+-----+-----+-----+-----+

```

Δημιουργία self-service network

Demo credentials

- root@Controller:~# . demo-openrc

Δημιουργία network

- root@Controller:~# neutron net-create private-net1

Δημιουργία subnet στο network

Demo credentials

- root@Controller:~# . demo-openrc
- root@Controller:~# neutron subnet-create --name private-subnet1 --dns-nameserver 8.8.8.8 private-net1 10.0.0.0/24
- root@Controller:~# neutron subnet-create --name private-subnet1 --dns-nameserver 8.8.8.8 private-net1 15.0.0.0/24
- root@Controller:~# neutron subnet-create --name private-subnet1 --dns-nameserver 8.8.8.8 private-net1 20.0.0.0/24

DNS_RESOLVER: **8.8.8.8**

SELFSERVICE_NETWORK_GATEWAY: **20.0.0.1, 15.0.0.1, 10.0.0.1**

SELFSERVICE_NETWORK_CIDR: **20.0.0.0/24, 15.0.0.0/24, 10.0.0.0/24**

Δημιουργία router

Admin credentials

- root@Controller:~# . admin-openrc
- root@Controller:~# neutron router-create router1

```
root@Controller:~# openstack router list
```

ID	Name	Status	State	Distributed	HA	Project
933f22c5-0393-497e-a17c-f83cae299420	demo_router	ACTIVE	UP	False	False	d680cfd5345a4406983dfd7bc70a8992
e7a653d3-88c4-422e-be4c-35261308711c	router1	ACTIVE	UP	False	False	d680cfd5345a4406983dfd7bc70a8992

Πρόσθεση του self-service network subnet ως interface στο δρομολογητή (router)

- root@Controller:~# neutron router-interface-add router1 private-subnet1

Ορίζουμε gateway στο provider network στο δρομολογητή

- root@Controller:~# neutron router-gateway-set router1 public

4.14.1 Επαλήθευση λειτουργίας

- root@Controller:~# . admin-openrc
- root@Controller:~# ip netns

```
root@Controller:~# ip netns
qdhcp-b5b025fe-5b32-4023-9421-fb1ec9bd02ea (id: 3)
qrouter-e7a653d3-88c4-422e-be4c-35261308711c (id: 2)
qrouter-933f22c5-0393-497e-a17c-f83cae299420 (id: 1)
qdhcp-71e00a67-7d2e-4c08-86e9-ac489c589ef3 (id: 0)
```

Καταχωρούμε τις πόρτες στο router για να προσδιορίζουν τη gateway IP στο provider network

- root@Controller:~# neutron router-port-list router

Ping αυτή τη διεύθυνση IP από τον controller node ή οποιοδήποτε host στο physical provider network

- root@Controller:~# ip netns exec qdhcp-b5b025fe-5b32-4023-9421-fb1ec9bd02ea ping \-c 4 20.0.0.18

```
root@Controller:~# ip netns exec qdhcp-b5b025fe-5b32-4023-9421-fb1ec9bd02ea ping -c 4 20.0.0.18
PING 20.0.0.18 (20.0.0.18) 56(84) bytes of data.
64 bytes from 20.0.0.18: icmp_seq=1 ttl=64 time=0.851 ms
64 bytes from 20.0.0.18: icmp_seq=2 ttl=64 time=0.439 ms
64 bytes from 20.0.0.18: icmp_seq=3 ttl=64 time=0.515 ms
64 bytes from 20.0.0.18: icmp_seq=4 ttl=64 time=0.481 ms

--- 20.0.0.18 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2999ms
rtt min/avg/max/mdev = 0.439/0.571/0.851/0.165 ms
```

Δημιουργία m1.nano flavor

- root@Controller:~# openstack flavor create --id 0 --vcpus 1 --ram 64 --disk 1 m1.nano
- root@Controller:~# openstack flavor list

```
root@Controller:~# openstack flavor list
```

ID	Name	RAM	Disk	Ephemeral	VCPUs	Is Public
0	m1.nano	64	1	0	1	True
c485dafc-cc9e-41cb-8964-519a73cc586a	demo_flavor	2048	25	0	3	False
ef09a74c-ba5d-42bf-bcd8-21748c5780b6	m1.medium	4096	30	0	2	True

Demo credentials

- root@Controller:~# . demo-openrc

Generate and add a key pair

- root@Controller:~# ssh-keygen -q -N ""
- root@Controller:~# openstack keypair create --public-key /root/.ssh/id_rsa.pub demo
- root@Controller:~# openstack keypair list

```
root@Controller:~# . admin-openrc
root@Controller:~# openstack keypair list
```

Name	Fingerprint
demo	65:e4:7f:89:49:fa:af:ab:cd:af:69:8a:d2:10:02:d3
Demo-kmarinis	0b:21:b0:86:96:15:17:a0:15:99:0e:5f:f4:7a:48:03
Demo-private	43:7a:f8:e7:f0:d3:e9:4a:c7:f7:fb:27:ed:ad:71:96

Add security group rules

Add rules to the default security group

- root@Controller:~# openstack security group rule create --proto icmp default

Permit secure shell (SSH) access

- root@Controller:~# openstack security group rule create --proto tcp --dst-port 22 default

4.14.2 Launch the instance on the provider network

Demo credentials

- root@Controller:~# . demo-openrc
- root@Controller:~# openstack server create --flavor m1.nano --image cirros --nic \ net-id=private-net1 --security-group default --key-name demo test-vm5

```
root@Controller:~# openstack server create --flavor m1.nano --image cirros --nic net-id=demo-2 --security-group default --key-name demo test-vm5
```

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	
OS-EXT-STS:power_state	NOSTATE
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	None
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	
adminPass	dw5WJR2zkzQK
config_drive	
created	2017-10-06T20:00:29Z
flavor	m1.nano (0)
hostId	
id	655828b7-34bc-4813-9a81-e433dcaa43e6
image	cirros (626b13af-f75b-4e79-a9f4-0d7b201cb435)
key_name	demo
name	test-vm5
progress	0
project_id	d680cfd5345a4406983fdf7bc70a8992
properties	
security_groups	name='default'
status	BUILD
updated	2017-10-06T20:00:29Z
user_id	42926646ba9e44f78821909af39ac946
volumes_attached	

- root@Controller:~# openstack server create --flavor m1.nano --image cirros --nic \ net-id=private-net1 --security-group default --key-name demo test-vm6

```
root@Controller:~# openstack server create --flavor m1.nano --image cirros --nic net-id=demo-2 --security-group default --key-name demo test-vm6
```

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	
OS-EXT-STS:power_state	NOSTATE
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	None
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	
adminPass	je6VX5wAUSTY
config_drive	
created	2017-10-06T20:01:04Z
flavor	m1.nano (0)
hostId	
id	d784ccf8-0973-42d0-8978-d71adaf9f5aa
image	cirros (626b13af-f75b-4e79-a9f4-0d7b201cb435)
key_name	demo
name	test-vm6
progress	0
project_id	d680cfd5345a4406983fdf7bc70a8992
properties	
security_groups	name='default'
status	BUILD
updated	2017-10-06T20:01:05Z
user_id	42926646ba9e44f78821909af39ac946
volumes_attached	

4.14.3 Status of instance

➤ root@Controller:~# openstack server list

```
root@Controller:~# openstack server list
```

ID	Name	Status	Networks	Image Name
d784ccf8-0973-42d0-8978-d71adaf9f5aa	test-vm6	ACTIVE	demo-2=20.0.0.15	cirros
655828b7-34bc-4813-9a81-e433dcaa43e6	test-vm5	ACTIVE	demo-2=20.0.0.18	cirros
44672c9c-0767-44f9-ab43-1ad0b1bd20e8	test-vm4	ACTIVE	demo=15.0.0.7	cirros
09362cb5-1a75-4c84-ae3d-dd98199899dd	test-vm3	ACTIVE	demo=15.0.0.11	cirros
795dc729-1a8c-4718-be57-796ffa0548e4	test-vm2	ACTIVE	private-net1=10.0.0.13	cirros
aca80c26-cbfa-4fc5-9019-2a8d2ce3aab5	test-vm1	ACTIVE	private-net1=10.0.0.6	cirros

Instances

Instance ID = Filter Launch Instance Delete Instances More Actions

Displaying 6 items

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
test-vm6	cirros	20.0.0.15	m1.nano	demo	Active	nova	None	Running	1 hour, 17 minutes	Create Snapshot
test-vm5	cirros	20.0.0.18	m1.nano	demo	Active	nova	None	Running	1 hour, 17 minutes	Create Snapshot
test-vm4	cirros	15.0.0.7	m1.nano	demo	Active	nova	None	Running	1 hour, 19 minutes	Create Snapshot
test-vm3	cirros	15.0.0.11	m1.nano	demo	Active	nova	None	Running	1 hour, 19 minutes	Create Snapshot
test-vm2	cirros	10.0.0.13	m1.nano	demo	Active	nova	None	Running	2 days, 5 hours	Create Snapshot
test-vm1	cirros	10.0.0.6	m1.nano	demo	Active	nova	None	Running	2 days, 5 hours	Create Snapshot

Displaying 6 items

Figure 16: Instances

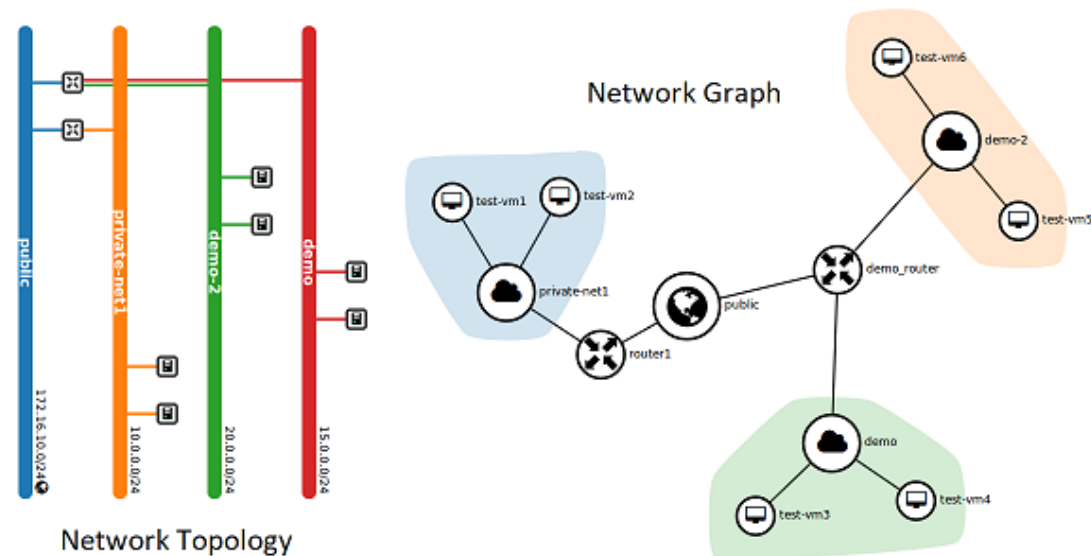


Figure 17: Network Topology & Graph

Κάνουμε είσοδο στο image που έχουμε εγκαταστήσει, στην περίπτωση μας το CIRROS για να επαληθεύσουμε την σωστή λειτουργία του και με την εντολή “ring”.

username: **cirros**
password: **cubswin:)**

```
login as 'cirros' user, default password: 'cubswin:)', use 'sudo' for root.
test-vm1 login: cirros
Password:
$
$
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:97:54:FD
          inet addr:10.0.0.6  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe97:54fd/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:68 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7687 (7.5 KiB)  TX bytes:10530 (10.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$
```

```
UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
RX packets:68 errors:0 dropped:0 overruns:0 frame:0
TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:7687 (7.5 KiB)  TX bytes:10530 (10.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$ ping -c 3 10.0.0.13
PING 10.0.0.13 (10.0.0.13): 56 data bytes
64 bytes from 10.0.0.13: seq=0 ttl=64 time=0.356 ms
64 bytes from 10.0.0.13: seq=1 ttl=64 time=0.206 ms
64 bytes from 10.0.0.13: seq=2 ttl=64 time=0.244 ms

--- 10.0.0.13 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.206/0.268/0.356 ms
$
```

Figure 18: test-vm1

```

login as 'cirros' user, default password: 'cubswin:)', use 'sudo' for root.
test-vm2 login: cirros
Password:
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:6D:B6:8B
          inet addr:10.0.0.13  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe6d:b68b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:73 errors:0 dropped:0 overruns:0 frame:0
          TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:8531 (8.3 KiB)  TX bytes:10530 (10.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$
UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
RX packets:73 errors:0 dropped:0 overruns:0 frame:0
TX packets:105 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:8531 (8.3 KiB)  TX bytes:10530 (10.2 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$ ping -c 3 10.0.0.6
PING 10.0.0.6 (10.0.0.6): 56 data bytes
64 bytes from 10.0.0.6: seq=0 ttl=64 time=0.297 ms
64 bytes from 10.0.0.6: seq=1 ttl=64 time=0.247 ms
64 bytes from 10.0.0.6: seq=2 ttl=64 time=0.235 ms

--- 10.0.0.6 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.235/0.259/0.297 ms
$

```

Figure 19: test-vm2

4.14 Object Storage service

Το OpenStack Object Storage είναι ένα multi-tenant object storage σύστημα. Είναι εξαιρετικά επεκτάσιμο και μπορεί να διαχειριστεί μεγάλες ποσότητες μη δομημένων δεδομένων με χαμηλό κόστος μέσω ενός API RESTful HTTP. (guide, n.d.)

Περιλαμβάνει τα ακόλουθα στοιχεία

Proxy servers (swift-proxy-server)

Αποδέχεται API object storage OpenStack και ακατέργαστα αιτήματα HTTP για τη μεταφόρτωση αρχείων, την τροποποίηση των μεταδεδομένων και τη δημιουργία container. Επίσης, εξυπηρετεί λίστες αρχείων ή container σε προγράμματα περιήγησης ιστού. Για να βελτιωθεί η απόδοση, ο διακομιστής μεσολάβησης μπορεί να χρησιμοποιήσει μια προαιρετική προσωρινή μνήμη που συνήθως αναπτύσσεται με memcache. (guide, n.d.)

Account servers (swift-account-server)

Διαχειρίζεται λογαριασμούς που ορίζονται με την αποθήκευση αντικειμένων (Object Storage).

Container servers (swift-container-server)

Διαχειρίζεται τη χαρτογράφηση των δοχείων ή των φακέλων, μέσα στο Storage Object.

Object servers (swift-object-server)

Διαχειρίζεται πραγματικά αντικείμενα, όπως αρχεία, στους κόμβους αποθήκευσης.

Various periodic processes

Εκτελεί εργασίες καθαρισμού στο μεγάλο κατάστημα δεδομένων. Οι υπηρεσίες αναπαραγωγής εξασφαλίζουν συνέπεια και διαθεσιμότητα μέσω του συμπλέγματος. Άλλες περιοδικές διαδικασίες περιλαμβάνουν τους ελεγκτές, τους ενημερωτές και τους εκκολλητούς.

WSGI middleware

Διαχειρίζεται τον έλεγχο ταυτότητας και συνήθως είναι ταυτότητα OpenStack.

swift client

Επιτρέπει στους χρήστες να υποβάλλουν εντολές στο API REST μέσω ενός πελάτη γραμμής εντολών εξουσιοδοτημένου admin user, reseller user, or swift user.

swift-init

Το σενάριο που προετοιμάζει το κτίριο του αρχείου του δακτυλίου, παίρνει ονόματα δαίμονα ως παράμετρο και προσφέρει εντολές.

swift-recon

Ένα εργαλείο cli που χρησιμοποιείται για την ανάκτηση διαφόρων μετρήσεων και πληροφοριών τηλεμετρίας για ένα σύμπλεγμα που έχει συγκεντρωθεί από το μεσαίο λογισμικό άμεσης επανεξέτασης.

swift-ring-builder

Δημιουργία δακτυλιδιού αποθήκευσης και χρησιμότητα ανακύκλωσης. (guide, n.d.)

4.14.1 Εγκατάσταση και διαμόρφωση - Controller node

Admin credentials

➤ `root@Controller:~# . admin-openrc`

Δημιουργία swift user

➤ `root@Controller:~# openstack user create --domain default \
--password-prompt swift`

```

root@Controller:~# . admin-openrc
root@Controller:~# openstack user create --domain default --password-prompt swift
User Password:
Repeat User Password:
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| domain_id     | default                                  |
| enabled       | True                                     |
| id            | 6e7f0a79a41f4493937cfc4d2cb14192      |
| name          | swift                                    |
| options       | {}                                       |
| password_expires_at | None                                    |
+-----+-----+

```

Πρόσθεση admin role στον swift user

- root@Controller:~# openstack role add --project service --user swift admin

Δημιουργία swift service

- root@Controller:~# openstack service create --name swift \
 --description "OpenStack Object Storage" object-store

```

root@Controller:~# openstack role add --project service --user swift admin
root@Controller:~# openstack service create --name swift \
> --description "OpenStack Object Storage" object-store
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| description    | OpenStack Object Storage                |
| enabled       | True                                     |
| id            | adacb3f181a0493784dff134a9b121f6      |
| name          | swift                                    |
| type          | object-store                             |
+-----+-----+

```

Δημιουργία Object Storage service API endpoints

- root@Controller:~# openstack endpoint create --region RegionOne \
 object-store public http://controller:8080/v1/AUTH_%(tenant_id)s
- root@Controller:~# openstack endpoint create --region RegionOne \
 object-store internal http://controller:8080/v1/AUTH_%(tenant_id)s
- root@Controller:~# openstack endpoint create --region RegionOne \
 object-store admin http://controller:8080/v1

```

root@Controller:~# openstack endpoint create --region RegionOne \
> object-store public http://controller:8080/v1/AUTH_%(tenant_id)s
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | 42eece24870240639691f84915993063       |
| interface  | public                                   |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | adacb3f181a0493784dff134a9b121f6       |
| service_name | swift                                    |
| service_type | object-store                             |
| url        | http://controller:8080/v1/AUTH_%(tenant_id)s |
+-----+-----+
root@Controller:~# openstack endpoint create --region RegionOne \
> object-store internal http://controller:8080/v1/AUTH_%(tenant_id)s
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | eb50fbd79ef499fad44082f3c8c4fab         |
| interface  | internal                                   |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | adacb3f181a0493784dff134a9b121f6       |
| service_name | swift                                    |
| service_type | object-store                             |
| url        | http://controller:8080/v1/AUTH_%(tenant_id)s |
+-----+-----+
root@Controller:~# openstack endpoint create --region RegionOne \
> object-store admin http://controller:8080/v1
+-----+-----+
| Field      | Value                                     |
+-----+-----+
| enabled    | True                                     |
| id         | ddc6ad43c6cc439ba9699613e36c8989       |
| interface  | admin                                     |
| region     | RegionOne                               |
| region_id  | RegionOne                               |
| service_id | adacb3f181a0493784dff134a9b121f6       |
| service_name | swift                                    |
| service_type | object-store                             |
| url        | http://controller:8080/v1               |
+-----+-----+

```

4.14.2 Εγκατάσταση και διαμόρφωση components

Εγκατάσταση πακέτων

- root@Controller:~# apt-get install swift swift-proxy python-swiftclient \
 python-keystoneclient python-keystonemiddleware \
 memcached

Δημιουργία /etc/swift directory

- root@Controller:~# mkdir /etc/swift

Proxy service configuration file

- root@Controller:~# curl -o /etc/swift/proxy-server.conf \
 https://git.openstack.org/cgit/openstack/swift/plain/etc/proxy-server.conf-sample

```

root@Controller:~# mkdir /etc/swift
root@Controller:~# curl -o /etc/swift/proxy-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/proxy-server.conf-sample
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 35755  100 35755    0     0  20645    0  0:00:01  0:00:01 --:--:-- 20643

```

Παραμετροποίηση του /etc/swift/proxy-server.conf

➤ root@Controller:~# vim /etc/swift/proxy-server.conf

[DEFAULT]

...

bind_port = 8080

user = swift

swift_dir = /etc/swift

```

[DEFAULT]
# bind_ip = 0.0.0.0
bind_port = 8080
# bind_timeout = 30
# backlog = 4096
swift_dir = /etc/swift
user = swift

```

[pipeline:main]

...

pipeline = catch_errors gatekeeper healthcheck proxy-logging cache container_sync bulk ratelimit authtoken keystoneauth container-quotas account-quotas slo dlo versioned_writes proxy-logging proxy-server

```

[pipeline:main]
# This sample pipeline uses tempauth and is used for SAIO dev work and
# testing. See below for a pipeline using keystone.
# pipeline = catch_errors gatekeeper healthcheck proxy-logging cache container_sync bulk tempurl ratelimit tempauth copy container-quotas account-quotas
pipeline = catch_errors gatekeeper healthcheck proxy-logging cache container_sync bulk ratelimit authtoken keystoneauth container-quotas account-quotas

```

[app:proxy-server]

use = egg:swift#proxy

...

account_autocreate = True

```

[app:proxy-server]
use = egg:swift#proxy
# You can override the default log routing for this app here:
# If set to 'true' authorized accounts that do not yet exist within the Swift
# cluster will be automatically created.
account_autocreate = True

```

[filter:keystoneauth]

use = egg:swift#keystoneauth

...

operator_roles = admin,user

```

#
[filter:keystoneauth]
use = egg:swift#keystoneauth
# The reseller_prefix option lists account namespaces that this middleware is

```



```
# operator_roles = admin, swiftoperator
operator_roles = admin,user
#
```

```
[filter:authtoken]
paste.filter_factory = keystonemiddleware.auth_token:filter_factory
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = swift
password = SWIFT_PASS
delay_auth_decision = True
```

```
[filter:authtoken]
paste.filter_factory = keystonemiddleware.auth_token:filter_factory
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = swift
password = 123456
delay_auth_decision = True
```

SWIFT_PASS: 123456

```
[filter:cache]
use = egg:swift#memcache
...
memcache_servers = controller:11211
```

```
[filter:cache]
Use = egg:swift#memcache
# You can override the default log routing for this filter here:
# set log_name = cache
# set log_facility = LOG_LOCAL0
# set log_level = INFO
# set log_headers = false
# set log_address = /dev/log
#
# If not set here, the value for memcache_servers will be read from
# memcache.conf (see memcache.conf-sample) or lacking that file, it will
# default to the value below. You can specify multiple servers separated with
# commas, as in: 10.1.2.3:11211,10.1.2.4:11211 (IPv6 addresses must
# follow rfc3986 section-3.2.2, i.e. [::1]:11211)
memcache_servers = controller:11211
```

4.14.3 Εγκατάσταση και διαμόρφωση Storage node -ObjectStorage1, ObjectStorage2-

Εγκατάσταση πακέτων

- root@ObjectStorage1:~# apt-get install xfsprogs rsync
- root@ObjectStorage2:~# apt-get install xfsprogs rsync

Format the /dev/sdb and /dev/sdc devices as XFS

- root@ObjectStorage1:~# mkfs.xfs /dev/sdb
- root@ObjectStorage2:~# mkfs.xfs /dev/sdc

- root@ObjectStorage1:~# mkfs.xfs /dev/sdb
- root@ObjectStorage2:~# mkfs.xfs /dev/sdc

```

root@ObjectStorage1:~# mkfs.xfs /dev/sdb
meta-data=/dev/sdb          isize=512    agcount=4, agsize=327680 blks
                =                   sectsz=512   attr=2, projid32bit=1
                =                   crc=1        finobt=1, sparse=0
data                =                   bsize=4096  blocks=1310720, imaxpct=25
                =                   sunit=0     swidth=0 blks
naming             =version 2          bsize=4096  ascii-ci=0 ftype=1
log                =internal log      bsize=4096  blocks=2560, version=2
                =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime           =none              extsz=4096  blocks=0, rtextents=0
root@ObjectStorage1:~# mkfs.xfs /dev/sdc
meta-data=/dev/sdc          isize=512    agcount=4, agsize=327680 blks
                =                   sectsz=512   attr=2, projid32bit=1
                =                   crc=1        finobt=1, sparse=0
data                =                   bsize=4096  blocks=1310720, imaxpct=25
                =                   sunit=0     swidth=0 blks
naming             =version 2          bsize=4096  ascii-ci=0 ftype=1
log                =internal log      bsize=4096  blocks=2560, version=2
                =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime           =none              extsz=4096  blocks=0, rtextents=0
root@ObjectStorage1:~# █
  
```

Create the mount point directory structure

- root@ObjectStorage1:~# mkdir -p /srv/node/sdb
- root@ObjectStorage2:~# mkdir -p /srv/node/sdc

Παραμετροποίηση του /etc/fstab

- root@ObjectStorage1:~# vim /etc/fstab
- root@ObjectStorage2:~# vim /etc/fstab

/dev/sdb /srv/node/sdb xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

/dev/sdc /srv/node/sdc xfs noatime,nodiratime,nobarrier,logbufs=8 0 2

```

# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>          <dump> <pass>
# / was on /dev/sda1 during installation
UUID=85eb0993-4a63-4bd3-a57f-73cf98d7f829 /                   ext4      errors=remount-ro 0    1
# swap was on /dev/sda5 during installation
UUID=b1bebe70-0684-49ff-ab89-703d3b2ad7c4 none                swap      sw                0    0
/dev/fd0          /media/floppy0     auto      rw,user,noauto,exec,utf8 0    0
/dev/sdb /srv/node/sdb xfs noatime,nodiratime,nobarrier,logbufs=8 0 2
/dev/sdc /srv/node/sdc xfs noatime,nodiratime,nobarrier,logbufs=8 0 2
  
```

Mount the devices

- root@ObjectStorage1:~# mount /srv/node/sdb
- root@ObjectStorage2:~# mount /srv/node/sdc

Δημιουργία ή παραμετροποίηση του `etc/rsyncd.conf` προσθέτοντας τα παρακάτω

- root@ObjectStorage1:~# vim etc/rsyncd.conf
- root@ObjectStorage2:~# vim etc/rsyncd.conf

```
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
address = MANAGEMENT_INTERFACE_IP_ADDRESS
```

```
[account]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/account.lock
```

```
[container]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/container.lock
```

```
[object]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/object.lock
```

```
uid = swift
gid = swift
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid
address = 172.16.10.234

[account]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/account.lock

[container]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/container.lock

[object]
max connections = 2
path = /srv/node/
read only = False
lock file = /var/lock/object.lock
```

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of ObjectStorage1: 172.16.10.234

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of ObjectStorage2: 172.16.10.235

Παραμετροποίηση του /etc/default/rsync

- root@ObjectStorage1:~# vim /etc/default/rsync
- root@ObjectStorage2:~# vim /etc/default/rsync

RSYNC_ENABLE = True

```
# defaults file for rsync daemon mode
# start rsync in daemon mode from init.d script?
# only allowed values are "true", "false", and "inetd"
# Use "inetd" if you want to start the rsyncd from inetd,
# all this does is prevent the init.d script from printing a message
# about not starting rsyncd (you still need to modify inetd's config yourself).
RSYNC_ENABLE=True
```

✓ Start the rsync service

- root@ObjectStorage1:~# service rsync start
- root@ObjectStorage2:~# service rsync start

4.14.4 Εγκατάσταση και διαμόρφωση components

Εγκατάσταση πακέτων

- root@ObjectStorage1:~# apt-get install swift swift-account swift-container \ swift-object

- root@ObjectStorage2:~# apt-get install swift swift-account swift-container \ swift-object

Obtain the accounting, container, and object service configuration files from the Object Storage source repository

- root@ObjectStorage1:~# curl -o /etc/swift/account-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/account-server.conf-sample
- root@ObjectStorage1:~# curl -o /etc/swift/container-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/container-server.conf-sample
- root@ObjectStorage1:~# curl -o /etc/swift/object-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/object-server.conf-sample
- root@ObjectStorage2:~# curl -o /etc/swift/account-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/account-server.conf-sample
- root@ObjectStorage2:~# curl -o /etc/swift/container-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/container-server.conf-sample
- root@ObjectStorage2:~# curl -o /etc/swift/object-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/object-server.conf-sample

```

root@ObjectStorage1:/etc/swift# curl -o /etc/swift/account-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/account-server.conf-sample
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 9480 100 9480 0 0 5480 0 0:00:01 0:00:01 --:--:-- 5482
root@ObjectStorage1:/etc/swift# vim /etc/swift/account-server.conf
root@ObjectStorage1:/etc/swift# curl -o /etc/swift/container-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/container-server.conf-sample
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 11028 100 11028 0 0 6546 0 0:00:01 0:00:01 --:--:-- 6544
root@ObjectStorage1:/etc/swift# vim /etc/swift/container-server.conf
root@ObjectStorage1:/etc/swift# curl -o /etc/swift/object-server.conf https://git.openstack.org/cgit/openstack/swift/plain/etc/object-server.conf-sample
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 17888 100 17888 0 0 10466 0 0:00:01 0:00:01 --:--:-- 10473
root@ObjectStorage1:/etc/swift#
  
```

Παραμετροποίηση του /etc/swift/account-server.conf

- root@ObjectStorage1:~# vim /etc/swift/account-server.conf
- root@ObjectStorage2:~# vim /etc/swift/account-server.conf

[DEFAULT]

...

bind_ip = MANAGEMENT_INTERFACE_IP_ADDRESS

bind_port = 6202

user = swift

swift_dir = /etc/swift

devices = /srv/node

mount_check = True

```
[DEFAULT]
bind_ip = 172.16.10.234
bind_port = 6202
# bind_timeout = 30
# backlog = 4096
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = true
```

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of BlockStorage1: **172.16.10.234**
MANAGEMENT_INTERFACE_IP_ADDRESS - IP of BlockStorage2: **172.16.10.235**

```
[pipeline:main]
pipeline = healthcheck recon account-server
```

```
[pipeline:main]
pipeline = healthcheck recon account-server

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

```
[filter:recon]
use = egg:swift#recon
...
recon_cache_path = /var/cache/swift
```

```
[filter:recon]
use = egg:swift#recon
recon_cache_path = /var/cache/swift
```

Παραμετροποίηση του /etc/swift/container-server.conf

- root@ObjectStorage1:~# vim /etc/swift/container-server.conf
- root@ObjectStorage2:~# vim /etc/swift/container-server.conf

```
[DEFAULT]
...
bind_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
bind_port = 6201
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = True
```

```
[DEFAULT]
bind_ip = 172.16.10.234
bind_port = 6200
# bind_timeout = 30
# backlog = 4096
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = true
```

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of BlockStorage1: **172.16.10.234**

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of BlockStorage2: **172.16.10.235**

```
[pipeline:main]
pipeline = healthcheck recon container-server
```

```
[pipeline:main]
pipeline = healthcheck recon object-server

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

```
[filter:recon]
use = egg:swift#recon
...
recon_cache_path = /var/cache/swift
```

```
[filter:recon]
use = egg:swift#recon
recon_cache_path = /var/cache/swift
#recon_lock_path = /var/lock
```

Παραμετροποίηση του /etc/swift/object-server.conf

- root@ObjectStorage1:~# vim /etc/swift/object-server.conf
- root@ObjectStorage2:~# vim /etc/swift/object-server.conf

```
[DEFAULT]
...
bind_ip = MANAGEMENT_INTERFACE_IP_ADDRESS
bind_port = 6200
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = True
```

```
[DEFAULT]
bind_ip = 172.16.10.234
bind_port = 6200
# bind_timeout = 30
# backlog = 4096
user = swift
swift_dir = /etc/swift
devices = /srv/node
mount_check = true
```

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of BlockStorage1: **172.16.10.234**

MANAGEMENT_INTERFACE_IP_ADDRESS - IP of BlockStorage2: **172.16.10.235**

```
[pipeline:main]
pipeline = healthcheck recon object-server
```



```
[pipeline:main]
pipeline = healthcheck recon object-server

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

```
[filter:recon]
use = egg:swift#recon
...
recon_cache_path = /var/cache/swift
recon_lock_path = /var/lock
```

```
[filter:recon]
use = egg:swift#recon
recon_cache_path = /var/cache/swift
recon_lock_path = /var/lock
```

Βεβαιωνόμαστε ότι έχουμε τα σωστά δικαιώματα ιδιοκτησίας της δομής καταλόγου σημείων προσάρτησης

- root@ObjectStorage1:~# chown -R swift:swift /srv/node
- root@ObjectStorage2:~# chown -R swift:swift /srv/node

Δημιουργία recon directory και διασφάλιση με τα κατάλληλα δικαιώματα

- root@ObjectStorage1:~# mkdir -p /var/cache/swift
- root@ObjectStorage1:~# chown -R root:swift /var/cache/swift
- root@ObjectStorage1:~# chmod -R 775 /var/cache/swift
- root@ObjectStorage2:~# mkdir -p /var/cache/swift
- root@ObjectStorage2:~# chown -R root:swift /var/cache/swift
- root@ObjectStorage2:~# chmod -R 775 /var/cache/swift

4.14.5 Δημιουργία και διανομή initial rings

Δημιουργία account ring

- root@Controller:~# cd /etc/swift

Δημιουργία φακέλου **account.builder**

- root@Controller:~# swift-ring-builder account.builder create 10 3 1

Πρόσθεση όλων των κόμβων στο ring

- root@Controller:~# swift-ring-builder /etc/swift/account.builder add r1z1-172.16.10.234:6202/sdb 100
- root@Controller:~# swift-ring-builder /etc/swift/account.builder add r1z1-172.16.10.234:6202/sdc 100

- root@Controller:~# swift-ring-builder /etc/swift/account.builder add r1z1-172.16.10.235:6202/sdb 100
- root@Controller:~# swift-ring-builder /etc/swift/account.builder add r1z1-172.16.10.235:6202/sdc 100
- root@Controller:~# swift-ring-builder account.builder

```
root@Controller:~# cd /etc/swift/
root@Controller:/etc/swift# swift-ring-builder account.builder
account.builder, build version 5
1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 4 devices, 0.00 balance, 0.00 dispersion
The minimum number of hours before a partition can be reassigned is 1 (0:00:00 remaining)
The overload factor is 0.00% (0.000000)
Ring file account.ring.gz is up-to-date
Devices:  id region zone  ip address:port replication ip:port  name weight partitions balance flags meta
         0      1      1 172.16.10.234:6202 172.16.10.234:6202  sdb 100.00      768    0.00
         1      1      1 172.16.10.234:6202 172.16.10.234:6202  sdc 100.00      768    0.00
         2      1      1 172.16.10.235:6202 172.16.10.235:6202  sdb 100.00      768    0.00
         3      1      1 172.16.10.235:6202 172.16.10.235:6202  sdc 100.00      768    0.00
```

- root@Controller:~# swift-ring-builder account.builder rebalance

Δημιουργία container ring

- root@Controller:~# swift-ring-builder container.builder create 10 3 1
- root@Controller:~# swift-ring-builder /etc/swift/container.builder add r1z1-172.16.10.234:6201/sdb 100
- root@Controller:~# swift-ring-builder /etc/swift/container.builder add r1z1-172.16.10.234:6201/sdc 100
- root@Controller:~# swift-ring-builder /etc/swift/container.builder add r1z1-172.16.10.235:6201/sdb 100
- root@Controller:~# swift-ring-builder /etc/swift/container.builder add r1z1-172.16.10.235:6201/sdc 100
- root@Controller:~# swift-ring-builder container.builder

```
root@Controller:/etc/swift# swift-ring-builder container.builder
container.builder, build version 5
1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 4 devices, 0.00 balance, 0.00 dispersion
The minimum number of hours before a partition can be reassigned is 1 (0:00:00 remaining)
The overload factor is 0.00% (0.000000)
Ring file container.ring.gz is up-to-date
Devices:  id region zone  ip address:port replication ip:port  name weight partitions balance flags meta
         0      1      1 172.16.10.234:6201 172.16.10.234:6201  sdb 100.00      768    0.00
         1      1      1 172.16.10.234:6201 172.16.10.234:6201  sdc 100.00      768    0.00
         2      1      1 172.16.10.235:6201 172.16.10.235:6201  sdb 100.00      768    0.00
         3      1      1 172.16.10.235:6201 172.16.10.235:6201  sdc 100.00      768    0.00
```

- root@Controller:~# swift-ring-builder container.builder rebalance

Δημιουργία object ring

- root@Controller:~# swift-ring-builder object.builder create 10 3 1
- root@Controller:~# swift-ring-builder /etc/swift/object.builder add r1z1-172.16.10.234:6200/sdb 100
- root@Controller:~# swift-ring-builder /etc/swift/object.builder add r1z1-172.16.10.234:6200/sdc 100
- root@Controller:~# swift-ring-builder /etc/swift/object.builder add r1z1-172.16.10.235:6200/sdb 100
- root@Controller:~# swift-ring-builder /etc/swift/object.builder add r1z1-172.16.10.235:6200/sdc 100
- root@Controller:~# swift-ring-builder object.builder

```
root@Controller:/etc/swift# swift-ring-builder object.builder
object.builder, build version 5
1024 partitions, 3.000000 replicas, 1 regions, 1 zones, 4 devices, 0.00 balance, 0.00 dispersion
The minimum number of hours before a partition can be reassigned is 1 (0:00:00 remaining)
The overload factor is 0.00% (0.000000)
Ring file object.ring.gz is up-to-date
Devices:  id region zone  ip address:port replication ip:port  name weight partitions balance flags meta
          0      1      1 172.16.10.234:6200 172.16.10.234:6200  sdb 100.00      768    0.00
          1      1      1 172.16.10.234:6200 172.16.10.234:6200  sdc 100.00      768    0.00
          2      1      1 172.16.10.235:6200 172.16.10.235:6200  sdb 100.00      768    0.00
          3      1      1 172.16.10.235:6200 172.16.10.235:6200  sdc 100.00      768    0.00
root@Controller:/etc/swift#
```

- root@Controller:~# swift-ring-builder object.builder rebalance

Εγκατάσταση openssh-server σε όλους τους κόμβους

- root@Controller:~# apt-get install openssh-server
- root@ObjectStorage1:~# apt-get install openssh-server
- root@ObjectStorage2:~# apt-get install openssh-server

Δικαιώματα στον φάκελο /etc/swift στα Object nodes

- root@ObjectStorage1:~# chown administrator /etc/swift
- root@ObjectStorage2:~# chown administrator /etc/swift

Αντιγραφή ring configuration files

- root@Controller:~# scp account.ring.gz container.ring.gz object.ring.gz administrator@ObjectStorage1:/etc/swift
- root@Controller:~# scp account.ring.gz container.ring.gz object.ring.gz administrator@ObjectStorage2:/etc/swift

Ανάκτηση του αρχείου /etc/swift/swift.conf από το Object Storage repository

- root@Controller:~# curl -o /etc/swift/swift.conf \
https://git.openstack.org/cgit/openstack/swift/plain/etc/swift.conf-sample

Παραμετροποίηση του αρχείου /etc/swift/swift.conf

- root@Controller:~# vim /etc/swift/swift.conf

[swift-hash]

...

swift_hash_path_suffix = HASH_PATH_SUFFIX

swift_hash_path_prefix = HASH_PATH_PREFIX

```
[swift-hash]
# swift_hash_path_suffix and swift_hash_path_prefix are used as part of the
# hashing algorithm when determining data placement in the cluster.
# These values should remain secret and MUST NOT change
# once a cluster has been deployed.
# Use only printable chars (python -c "import string; print(string.printable)")
swift_hash_path_suffix = kostas
swift_hash_path_prefix = marinis
```

- ✓ Αντικατάσταση των HASH_PATH_PREFIX και HASH_PATH_SUFFIX με μοναδικά values

[storage-policy:0]

...

name = Policy-0

default = yes

```
[storage-policy:0]
name = Policy-0
default = yes
#policy_type = replication
aliases = yellow, orange
```

Αντιγραφή του αρχείου **swift.conf** στον φάκελο /etc/swift στο κάθε ένα Object node

- root@Controller:~# scp swift.conf administrator@ObjectStorage1:/etc/swift
- root@Controller:~# scp swift.conf administrator@ObjectStorage2:/etc/swift

Αλλαγή δικαιωμάτων σε όλους τους κόμβους

- root@Controller:~# chown -R root:swift /etc/swift
- root@ObjectStorage1:~# chown -R root:swift /etc/swift
- root@ObjectStorage2:~# chown -R root:swift /etc/swift

Επανεκκίνηση των services στον Controller node

- root@Controller:~# service memcached restart
- root@Controller:~# service swift-proxy restart

✓ **Εκκίνηση** του service swift-init στα ObjectStorage node

- root@ObjectStorage1:~# swift-init all start
- root@ObjectStorage2:~# swift-init all start

4.14.6 Επαλήθευση λειτουργίας

Demo credentials

- root@Controller:~# . demo-openrc

service status

- root@Controller:~# swift stat

```
root@Controller:~# . demo-openrc
root@Controller:~# swift stat
Account: AUTH_e2c082fd3fd04b41aa84f90a94208ce8
Containers: 3
Objects: 0
Bytes: 0
Containers in policy "policy-0": 3
Objects in policy "policy-0": 0
Bytes in policy "policy-0": 0
X-Account-Project-Domain-Id: default
X-Openstack-Request-Id: txebc8425f8cd24b61890fe-0059a866d0
X-Timestamp: 1503740705.35088
X-Trans-Id: txebc8425f8cd24b61890fe-0059a866d0
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

Δημιουργία container1 και demo containers

- root@Controller:~# openstack container create container1
- root@Controller:~# openstack container create demo

```
root@Controller:~# openstack container create container1
+-----+-----+-----+
| account | container | x-trans-id |
+-----+-----+-----+
| AUTH_e2c082fd3fd04b41aa84f90a94208ce8 | container1 | txa2443595ce88432590954-0059a86712 |
+-----+-----+-----+
```

Δημιουργία testfile1 και testfile2

- root@Controller:~# vim testfile1
- root@Controller:~# vim testfile2

Upload test file -testfile1 και testfile2-

- root@Controller:~# openstack object create container1 testfile1
- root@Controller:~# openstack object create demo testfile2

```

root@Controller:~# vim testfile1
root@Controller:~#
root@Controller:~#
root@Controller:~# openstack object create container1 testfile1
+-----+-----+-----+
| object | container | etag |
+-----+-----+-----+
| testfile1 | container1 | 69089a812e10f18a460849767fb2973c |
+-----+-----+-----+
  
```

Περιεχόμενο container1 και demo

- root@Controller:~# openstack object list container1
- root@Controller:~# openstack object list demo

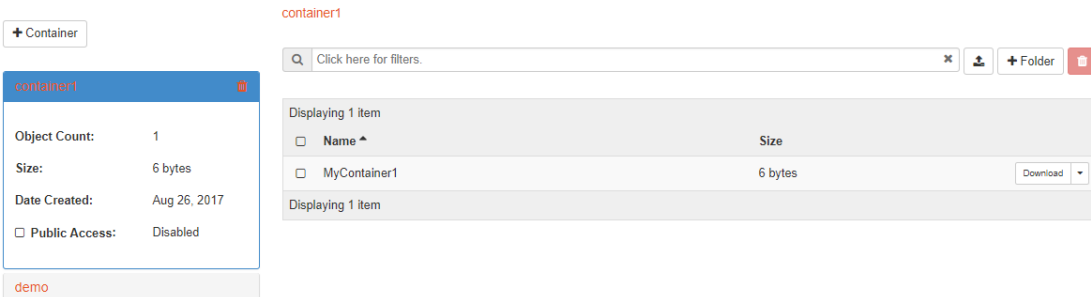
Download test file -testfile1 και testfile2-

- root@Controller:~# openstack object save container1 testfile1
- root@Controller:~# openstack object save demo testfile2

```

root@Controller:~# openstack object list container1
+-----+
| Name |
+-----+
| testfile1 |
+-----+
root@Controller:~# openstack object save container1 testfile1
root@Controller:~#
  
```

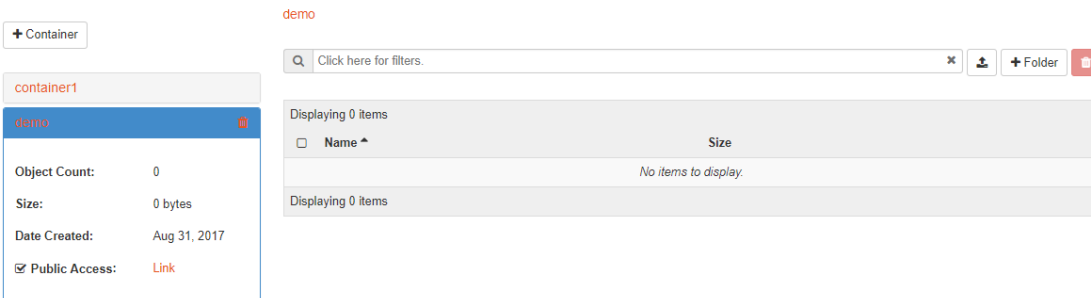
Containers



The screenshot shows the OpenStack Object Storage interface for a container named 'container1'. On the left, a sidebar displays container statistics: Object Count: 1, Size: 6 bytes, Date Created: Aug 26, 2017, and Public Access: Disabled. The main area shows a search bar and a table with one object: 'MyContainer1' (6 bytes). A 'Download' button is visible next to the object name.

Figure 20: Container1

Containers



The screenshot shows the OpenStack Object Storage interface for a container named 'demo'. The sidebar displays container statistics: Object Count: 0, Size: 0 bytes, Date Created: Aug 31, 2017, and Public Access: Link. The main area shows a search bar and an empty table with the message 'No items to display.'

Figure 21: Container Demo

4.15 Orchestration service

Το Orchestration service παραχωρεί ένα template-based orchestration για την περιγραφή ενός cloud service τρέχοντας το API του OpenStack ώστε να μπορούν να λειτουργήσουν οι εφαρμογές του cloud. Το λογισμικό ενσωματώνει και άλλες κύριες εφαρμογές του Openstack σε ένα one-file πρότυπο σύστημα. Το περίγραμμα αυτό επιτρέπει την δημιουργία των περισσότερων OpenStack resource τύπων, όπως για παράδειγμα, instances, floating IPs, volumes, security groups και χρήστες. Επιπλέον, δίνει την δυνατότητα προχωρημένων λειτουργιών όπως το υψηλή διαθεσιμότητα στο instance, instance auto-scaling και nested stacks. Αυτές οι λειτουργίες επιτρέπουν το OpenStack core project να μπορεί να εξυπηρετεί μεγάλο όγκο χρηστών.

Η υπηρεσία αυτή επιτρέπει τους deployers του συστήματος να το ενσωματώσουν με το Orchestration service άμεσα ή μέσω custom plug-ins. (guide, n.d.)

Το Orchestration service αποτελείται από τα ακόλουθα μέρη

heat command-line client

CLI που επικοινωνεί με το heat-api και τρέχει το AWS CloudFormation APIs. Οι τελικοί προγραμματιστές μπορούν άμεσα να χρησιμοποιήσουν το Orchestration REST API.

heat-api component

OpenStack-native REST API που επεξεργάζεται τα αιτήματα API στέλνοντας τα στο heat-engineer μέσω Remote Procedure Call (RPC).

heat-api-cfn component

Ένα API Query AP που είναι συμβατό με το CloudFormation του AWS. Επεξεργάζεται αιτήματα API στέλνοντάς τα στον θερμικό κινητήρα μέσω RPC.

heat-api-cloudwatch component

Μια υπηρεσία API τύπου CloudWatch για το heat project.

heat-engine

Ενορχηστρώνει την εκκίνηση προτύπων και παρέχει εκδηλώσεις πίσω στον καταναλωτή API.

4.15.1 Εγκατάσταση και διαμόρφωση

Προαπαιτούμενα

Δημιουργία βάσης δεδομένων heat

➤ root@Controller:~# mysql -u root -p

```
MariaDB [(none)]> CREATE DATABASE heat;
MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'localhost' \
  IDENTIFIED BY 'HEAT_DBPASS';
MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'%' \
  IDENTIFIED BY 'HEAT_DBPASS';
MariaDB [(none)]> exit
```



```

root@Controller:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 83
Server version: 10.0.31-MariaDB-0ubuntu0.16.04.2 Ubuntu 16.04

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE heat;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'localhost' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON heat.* TO 'heat'@'%' IDENTIFIED BY '123456';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit
Bye
root@Controller:~#
  
```

HEAT_DBPASS: 123456

Admin credentials

➤ root@Controller:~# . admin-openrc

Δημιουργία heat user

➤ root@Controller:~# openstack user create --domain default --password-prompt heat
 User Password: 123456

Πρόσθεση ρόλου admin στον heat user

➤ root@Controller:~# openstack role add --project service --user heat admin

```

root@Controller:~# . admin-openrc
root@Controller:~# openstack user create --domain default --password-prompt heat
User Password:
Repeat User Password:
+-----+-----+
| Field          | Value          |
+-----+-----+
| domain_id      | default        |
| enabled        | True           |
| id             | 3d15f03320df4877878e58fae7ce55bd |
| name           | heat           |
| options        | {}             |
| password_expires_at | None          |
+-----+-----+
root@Controller:~#
root@Controller:~#
root@Controller:~#
root@Controller:~# openstack role add --project service --user heat admin
root@Controller:~#
  
```

Δημιουργία heat και heat-cfn service

➤ root@Controller:~# openstack service create --name heat \
 --description "Orchestration" orchestration

```

root@Controller:~# openstack service create --name heat --description "Orchestration" orchestration
+-----+-----+
| Field | Value |
+-----+-----+
| description | Orchestration |
| enabled | True |
| id | c6f147556c37421e92ee29f727c16859 |
| name | heat |
| type | orchestration |
+-----+-----+

```

- root@Controller:~# openstack service create --name heat-cfn \
 --description "Orchestration" cloudformation

```

root@Controller:~# openstack service create --name heat-cfn --description "Orchestration" cloudformation
+-----+-----+
| Field | Value |
+-----+-----+
| description | Orchestration |
| enabled | True |
| id | a46c018f7e734bb9ae42f9cc4d0407de |
| name | heat-cfn |
| type | cloudformation |
+-----+-----+

```

4.15.2 Δημιουργία Orchestration service API endpoints

- root@Controller:~# openstack endpoint create --region RegionOne \
 orchestration public http://controller:8004/v1/%(tenant_id)s
- root@Controller:~# openstack endpoint create --region RegionOne \
 orchestration internal http://controller:8004/v1/%(tenant_id)s
- root@Controller:~# openstack endpoint create --region RegionOne \
 orchestration admin http://controller:8004/v1/%(tenant_id)s

```

root@Controller:~# openstack endpoint create --region RegionOne orchestration public http://controller:8004/v1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | a71a94f6f6134b438383b3da4e7cb83a |
| interface | public |
| region | RegionOne |
| region_id | RegionOne |
| service_id | c6f147556c37421e92ee29f727c16859 |
| service_name | heat |
| service_type | orchestration |
| url | http://controller:8004/v1/%(tenant_id)s |
+-----+-----+
root@Controller:~#
root@Controller:~#
root@Controller:~# openstack endpoint create --region RegionOne orchestration internal http://controller:8004/v1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 044f06a7a8bd44f98ff596f48df00e88 |
| interface | internal |
| region | RegionOne |
| region_id | RegionOne |
| service_id | c6f147556c37421e92ee29f727c16859 |
| service_name | heat |
| service_type | orchestration |
| url | http://controller:8004/v1/%(tenant_id)s |
+-----+-----+
root@Controller:~#
root@Controller:~# openstack endpoint create --region RegionOne orchestration admin http://controller:8004/v1/%(tenant_id)s
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | b5b3053810c94acd9555cae438f56bfc |
| interface | admin |
| region | RegionOne |
| region_id | RegionOne |
| service_id | c6f147556c37421e92ee29f727c16859 |
| service_name | heat |
| service_type | orchestration |
| url | http://controller:8004/v1/%(tenant_id)s |
+-----+-----+

```

- root@Controller:~# openstack endpoint create --region RegionOne \
 cloudformation public http://controller:8000/v1

- root@Controller:~# openstack endpoint create --region RegionOne \
 cloudformation internal http://controller:8000/v1
- root@Controller:~# openstack endpoint create --region RegionOne \
 cloudformation admin http://controller:8000/v1

```

root@Controller:~# openstack endpoint create --region RegionOne cloudformation public http://controller:8000/v1
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 1b9e6ca96d82469182465d9048ac4c20 |
| interface | public |
| region | RegionOne |
| region_id | RegionOne |
| service_id | a46c018f7e734bb9ae42f9cc4d0407de |
| service_name | heat-cfn |
| service_type | cloudformation |
| url | http://controller:8000/v1 |
+-----+-----+
root@Controller:~#
root@Controller:~#
root@Controller:~# openstack endpoint create --region RegionOne cloudformation internal http://controller:8000/v1
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | 434c487f4444446587e00856eec500d9 |
| interface | internal |
| region | RegionOne |
| region_id | RegionOne |
| service_id | a46c018f7e734bb9ae42f9cc4d0407de |
| service_name | heat-cfn |
| service_type | cloudformation |
| url | http://controller:8000/v1 |
+-----+-----+
root@Controller:~#
root@Controller:~# openstack endpoint create --region RegionOne cloudformation admin http://controller:8000/v1
+-----+-----+
| Field | Value |
+-----+-----+
| enabled | True |
| id | e5547ff5ea04416dacd176f51025bbfb |
| interface | admin |
| region | RegionOne |
| region_id | RegionOne |
| service_id | a46c018f7e734bb9ae42f9cc4d0407de |
| service_name | heat-cfn |
| service_type | cloudformation |
| url | http://controller:8000/v1 |
+-----+-----+

```

Δημιουργία **heat domain** ώστε να περιέχει projects και χρήστες για στοίβες (stacks)

- root@Controller:~# openstack domain create --description \
 "Stack projects and users" heat

```

root@Controller:~# openstack domain create --description "Stack projects and users" heat
+-----+-----+
| Field | Value |
+-----+-----+
| description | Stack projects and users |
| enabled | True |
| id | c1e5a7c0d79a454584b74a8ed8b1708f |
| name | heat |
+-----+-----+

```

Δημιουργία **heat_domain_admin user** ώστε να διαχειρίζεται projects και χρήστες στο heat domain

- root@Controller:~# openstack user create --domain heat --password-prompt \
 heat_domain_admin
 User Password: **123456**

```

root@Controller:~#
root@Controller:~# openstack user create --domain heat --password-prompt heat_domain_admin
User Password:
Repeat User Password:
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | c1e5a7c0d79a454584b74a8ed8b1708f |
| enabled | True |
| id | bd766005458e483a804efe7b854a1f6e |
| name | heat_domain_admin |
| options | {} |
| password_expires_at | None |
+-----+-----+

```

Admin role στον heat domain admin user στο domain heat

- root@Controller:~# openstack role add --domain heat --user-domain heat \
 --user heat_domain_admin admin

Δημιουργία heat_stack_owner role

- root@Controller:~# openstack role create heat_stack_owner

```

root@Controller:~# openstack role create heat_stack_owner
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | None |
| id | 14d19960afa84388915c9677ea799b14 |
| name | heat_stack_owner |
+-----+-----+

```

Δημιουργία heat stack owner role

- root@Controller:~# openstack role add --project demo --user demo \
 heat_stack_owner

```

root@Controller:~# openstack role add --project demo --user demo heat_stack_owner
root@Controller:~#
root@Controller:~# openstack role add --project demo --user admin heat_stack_owner
root@Controller:~#
root@Controller:~# openstack role add --project demo --user kmarinis heat_stack_owner
root@Controller:~#
root@Controller:~# openstack role add --project demo --user swift heat_stack_owner
root@Controller:~#
root@Controller:~# openstack role add --project demo --user heat heat_stack_owner

```

- ✓ Heat stack owner role σε όλους τους χρήστες που διαχειρίζονται στοίβες (stacks)

Δημιουργία heat stack user role

- root@Controller:~# openstack role create heat_stack_user

```

root@Controller:~# openstack role create heat_stack_user
+-----+-----+
| Field | Value |
+-----+-----+
| domain_id | None |
| id | 8aff9e75b8654f45b3b91b2c6f469f5a |
| name | heat_stack_user |
+-----+-----+

```

4.15.3 Εγκατάσταση και διαμόρφωση components

Εγκατάσταση πακέτων

➤ root@Controller:~# apt-get install heat-api heat-api-cfn heat-engine

Παραμετροποίηση του αρχείου /etc/heat/heat.conf

➤ root@Controller:~# vim /etc/heat/heat.conf

[DEFAULT]

...

rpc_backend = rabbitdemo

```
# Reason: Replaced by [DEFAULT]/transport_url
rpc_backend = rabbit
```

heat_metadata_server_url = http://controller:8000

heat_waitcondition_server_url = http://controller:8000/v1/waitcondition

```
# (string value)
heat_metadata_server_url = http://controller:8000

# URL of the Heat waitcondition server. (string value)
heat_waitcondition_server_url = http://controller:8000/v1/waitcondition
```

stack_domain_admin = heat_domain_admin

stack_domain_admin_password = HEAT_DOMAIN_PASS

stack_user_domain_name = heat

```
# Keystone domain name which contains heat template-defined users. If
# 'stack_user_domain_id' option is set, this option is ignored. (string value)
stack_user_domain_name = heat

# Keystone username, a user with roles sufficient to manage users and projects
# in the stack_user_domain. (string value)
stack_domain_admin = heat_domain_admin

# Keystone password for stack_domain_admin user. (string value)
stack_domain_admin_password = 123456
```

HEAT_DOMAIN_PASS: 123456

[database]

...

connection = mysql+pymysql://heat:HEAT_DBPASS@controller/heat

```
# Deprecated group/name - [sql]/connection
connection = mysql+pymysql://heat:123456@controller/heat
```

[oslo_messaging_rabbit]

...

rabbit_host = controller

rabbit_userid = openstack

rabbit_password = RABBIT_PASS

```
# Reason: Replaced by [DEFAULT]/transport_url
rabbit_host = controller

# Reason: Replaced by [DEFAULT]/transport_url
rabbit_userid = openstack

# DEPRECATED: The RabbitMQ password. (string value)
# Deprecate group/name - [DEFAULT]/rabbit_password
# This option is deprecated for removal.
# Its value may be silently ignored in the future.
# Reason: Replaced by [DEFAULT]/transport_url
rabbit_password = 123456
```

RABBIT_PASS: 123456
HEAT_DBPASS: 123456

[keystone_authtoken]

```
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = heat
password = HEAT_PASS
```

```
[keystone_authtoken]
#
# From keystone.middleware.auth_token
#
# Complete "public" Identity API endpoint. This endpoint should not be an
# "admin" endpoint, as it should be accessible by all end users.
# Unauthenticated clients are redirected to this endpoint to authenticate.
# Although this endpoint should ideally be unversioned, client support in the
# wild varies. If you're using a versioned v2 endpoint here, then this should
# *not* be the same endpoint the service user utilizes for validating tokens,
# because normal end users may not be able to reach that endpoint. (string
# value)
#auth_uri = <None>
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = heat
password = 123456
```

HEAT_PASS: 123456

[trustee]

```
...
auth_type = password
auth_url = http://controller:35357
username = heat
password = HEAT_PASS
user_domain_name = default
```

```
[trustee]
#
# From heat.common.context
#
# Authentication type to load (string value)
# Deprecated group/name - [trustee]/auth_plugin
auth_type = password
# Config Section from which to load plugin specific options (string value)
#auth_section = <None>
# Authentication URL (string value)
auth_url = http://controller:35357

# Username (string value)
# Deprecated group/name - [trustee]/user-name
username = heat

# User's domain id (string value)
#user_domain_id = <None>

# User's domain name (string value)
user_domain_name = default

# User's password (string value)
password = 123456
```

HEAT_PASS: 123456

[clients_keystone]

...

auth_uri = http://controller:35357

```
# Unversioned keystone url in format like http://0.0.0.0:5000. (string value)
auth_uri = http://controller:35357
```

[ec2authtoken]

...

auth_uri = http://controller:5000

```
[ec2authtoken]
#
# From heat.api.aws.ec2token
#
# Authentication Endpoint URI. (string value)
auth_uri = http://controller:5000
```

Populate the Orchestration database

➤ root@Controller:~# su -s /bin/sh -c "heat-manage db_sync" heat

4.15.4 Ολοκλήρωση εγκατάστασης

Επανεκκίνηση Orchestration services

➤ root@Controller:~# service heat-api restart
➤ root@Controller:~# service heat-api-cfn restart

- root@Controller:~# service heat-engine restart

4.15.5 Επαλήθευση λειτουργίας

Admin credentials

- root@Controller:~# . admin-openrc

List service components

- root@Controller:~# openstack orchestration service list

```

root@Controller:~# . admin-openrc
root@Controller:~# openstack orchestration service list
+-----+-----+-----+-----+-----+-----+-----+
| Hostname | Binary | Engine ID | Host | Topic | Updated At | Status |
+-----+-----+-----+-----+-----+-----+-----+
| Controller | heat-engine | b2c280cf-77cc-4663-b3bb-2115a0db5c07 | Controller | engine | 2017-08-29T16:16:31.000000 | up |
| Controller | heat-engine | e975ea4d-b796-4416-963a-6c349b0e5e9a | Controller | engine | 2017-08-29T16:16:30.000000 | up |
| Controller | heat-engine | be901c61-f841-4c00-9956-04754c2dc5f4 | Controller | engine | 2017-08-29T16:16:31.000000 | up |
| Controller | heat-engine | 2d7619d5-4fdd-42b3-841a-2cb626717f26 | Controller | engine | 2017-08-29T16:16:31.000000 | up |
| Controller | heat-engine | 775c2374-c760-4ef8-b306-4150118e4eea | Controller | engine | 2017-08-29T16:16:30.000000 | up |
| Controller | heat-engine | 330d0c55-b2bf-4e09-b51b-bc23008f99e6 | Controller | engine | 2017-08-29T16:16:30.000000 | up |
| Controller | heat-engine | 49c0a004-b933-4336-9bfe-2bb6d854e47d | Controller | engine | 2017-08-29T16:16:31.000000 | up |
| Controller | heat-engine | a922939d-556a-42e2-86ef-b6df8f8089ea | Controller | engine | 2017-08-29T16:16:30.000000 | up |
+-----+-----+-----+-----+-----+-----+-----+
root@Controller:~#
  
```

4.15.6 Δημιουργία stack

Δημιουργούμε ένα template όπως το παρακάτω:

heat_template_version: 2017-09-08

description: Template to create a Nova custom flavor.

resources:

nova_flavor:

type: OS::Nova::Flavor

properties:

ephemeral: 1

is_public: true

name: custom-tiny.m1

ram: 512

vcpus: 1

Admin credentials

- root@Controller:~# . admin-openrc

Δημιουργία demo stack

- root@Controller:~# openstack stack create -t demo-temp-simple.yaml stack

```

root@Controller:~# openstack stack create -t demo-temp-simple.yaml stack
+-----+-----+
| Field | Value |
+-----+-----+
| id | a967692a-bcba-41f0-ade6-b2208de3cbd3 |
| stack_name | stack |
| description | Template to create a Nova custom flavor. |
| creation_time | 2017-09-08T16:55:49Z |
| updated_time | None |
| stack_status | CREATE IN PROGRESS |
| stack_status_reason | Stack CREATE started |
+-----+-----+
  
```

Μετά από λίγα λεπτά μπορούμε να ελέγξουμε αν το stack έχει δημιουργηθεί επιτυχώς

➤ root@Controller:~# openstack stack list

```

root@Controller:~# openstack stack list
+-----+-----+-----+-----+-----+
| ID | Stack Name | Stack Status | Creation Time | Updated Time |
+-----+-----+-----+-----+-----+
| a967692a-bcba-41f0-ade6-b2208de3cbd3 | stack | CREATE_COMPLETE | 2017-09-08T16:55:49Z | None |
+-----+-----+-----+-----+-----+
  
```

Stack Name ▾

Filter
+ Launch Stack
Preview Stack
Delete Stacks

More Actions ▾

Displaying 1 item

<input type="checkbox"/> Stack Name	Created	Updated	Status	Actions
<input type="checkbox"/> stack	4 weeks	Never	Check Complete	Check Stack ▾

Displaying 1 item

Figure 22: Stack

stack
Check Stack ▾

Topology
Overview
Resources
Events
Template

stack

Check Complete

Figure 23: Stack Topology

Συμπέρασμα

OpenStack deployment και OpenStack development έχουν και τα δύο το δικό τους επίπεδο σπουδαιότητας, με διαφορετικές πολυπλοκότητες.

Για την ανάπτυξη του υπάρχουν αρκετοί τρόποι όπως βλέπουμε ενδεικτικά παρακάτω:

- Devstack (επιλογή προγραμματιστών)
- Packstack (για απόδειξη των εννοιών)
- Χειροκίνητη εγκατάσταση κάθε υπηρεσίας
- Αυτοματοποιημένα εργαλεία εγκατάστασης ή ανάπτυξης όπως OpenStack ansible, kolla (χρησιμοποιώντας containerized services)

Δεδομένου ότι για την υλοποίηση της εφαρμογής έγινε Χειροκίνητη εγκατάσταση κάθε υπηρεσίας σε περιβάλλον Ubuntu Server 16.04 LTS παρατηρήθηκαν αρκετά σφάλματα και επιλύθηκαν με τις παραμετροποιήσεις που θα παρατεθούν παρακάτω. Πιο συγκεκριμένα, δοκιμάστηκαν οι εκδόσεις Ubuntu Server 14.04 και Ubuntu Server 16.04 και για το λόγο του ότι παρατηρήθηκε μεγάλη αστάθεια στην πρώτη, προχωρήσαμε στην δεύτερη που θα την χαρακτηρίζαμε αρκετά σταθερή.

Ζήτημα 1

Στην επαλήθευση λειτουργίας του Dashboard, η αρχική σελίδα `http:controller/horizon` της εφαρμογής (GUI) δεν εμφανίστηκε με επιτυχία όπως αναμενόταν.

Για την επίλυση του σφάλματος ακολουθούμε τα παρακάτω βήματα:

Αλλαγή δικαιωμάτων στο **"secret_key"**

- `chown -R www-data:www-data /var/lib/openstack-dashboard/secret_key`
- `chmod 600 /var/lib/openstack-dashboard/secret_key`
- Πρόσθεση της γραμμής **"WSGIApplicationGroup %{GLOBAL}"** στο αρχείο `/etc/apache2/conf-available/openstack-dashboard.conf` όπως περιγράφεται και στην ενότητα 4.12.1.

Ζήτημα 2

Κατά την διαδικασία εγκατάστασης του object-storage (swift service) πρέπει να αποκτήσουμε το proxy service configuration file από το Object Storage source repository.

Αυτό που δίνετε σαν το ακριβές repository είναι το παρακάτω:

<https://git.openstack.org/cgit/openstack/swift/plain/etc/proxy-server.conf-sample?h=stable/mitaka>

στην συνέχεια όμως, δεν μπορούμε να παραμετροποιήσουμε το αρχείο `proxy-server.conf`, γιατί παρατηρούμε ότι έχει δημιουργήσει ένα αρχείο `html` και όχι το `config file` που χρειαζόμαστε.

Το σωστό repository για την έκδοση που εγκαθιστούμε ώστε να δημιουργήσει το σωστό config file είναι το παρακάτω:

```
https://git.openstack.org/cgit/openstack/swift/plain/etc/proxy-server.conf-sample?h=stable/newton
```

Ζήτημα 3

Αρχικά, κάθε OpenStack service εμφανιζόταν ενεργό. Ωστόσο, παρόλο, που όλες οι εγκατεστημένες υπηρεσίες ήταν ενεργοποιημένες και λειτουργούσαν, τα launched instances (OpenStack VMs) είχαν “κολλήσει” σε κατάσταση BUILD και δεν άλλαζαν την κατάσταση τους σε ACTIVE.

Μετά από προσεκτική ανάλυση, έγινε σωστή εκκίνηση των instances, ώστε να έρθουν σε κατάσταση ACTIVE διορθώνοντας τις διαμορφώσεις του nova service.

Για την επίλυση ακολουθήθηκαν τα παρακάτω βήματα:

- Δημιουργία Placement service user που χρησιμοποιεί το επιλεγμένο PLACEMENT_PASS
- Πρόσθεση του Placement user στο service project με admin role
- Δημιουργία Placement API entry in the service catalog
- Δημιουργία Placement API service endpoints
- Εγκατάσταση πακέτου nova-placement-api
- Παραμετροποίηση του αρχείου nova.conf στον Controller και στον Compute node
- Register the cell0 database
- Δημιουργία cell1 cell
- Discover compute hosts

Ζήτημα 4

Πριν από την εκκίνηση του instance, εντοπίσαμε ένα πρόβλημα με τα δίκτυα που δημιουργήθηκαν ήδη για το demo tenant.

Το public network που είχε δημιουργηθεί στο OpenStack ήταν διαφορετικό από το δίκτυο στο οποίο ανήκαν τα hosts. Έτσι αφαιρέσαμε πρώτα όλα αυτά τα ανεπιθύμητα δίκτυα και διαγράψαμε την τοπολογία του δικτύου. Στη συνέχεια δημιουργήθηκε ξανά.

Ζήτημα 5

Τώρα που τα δίκτυά ήταν έτοιμα στο OpenStack και τα launched instances έγιναν ACTIVE, δεν ήταν σε θέση να επικοινωνήσουν (ping) μεταξύ τους. Αναλύθηκαν και πάλι τα σχετικά αρχεία καταγραφής (logs), επαληθεύτηκαν και διορθώθηκαν οι διαμορφώσεις και στη συνέχεια έγινε ping αφού προστέθηκαν security group rules. Επίσης έγινε ping και από τον Controller node με χρήση της εντολής 'ip netns'.

+ Add Rule
Delete Rules

Displaying 4 items						
<input type="checkbox"/> Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Actions
<input type="checkbox"/> Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/> Egress	IPv4	ICMP	Any	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/> Ingress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule
<input type="checkbox"/> Egress	IPv4	TCP	1 - 65535	0.0.0.0/0	-	Delete Rule
Displaying 4 items						

Figure 24: Security Group Rules

Ζήτημα 6

Χρειάστηκε να γίνει resetting στα quota count. Υπήρξε λάθος τιμή στην Vcpu και αυτό μπορεί να οφείλεται στο γεγονός ότι παλαιότερες καταχωρήσεις των instances κόλλησαν σε κατάσταση BUILD. Η λύση αυτή προτείνεται για την έκδοση ICEHOUSE αλλά στην περίπτωση μας δούλεψε χωρίς πρόβλημα.

- `mysql -u nova -p<password> nova`
- `select * from quota_usages;`
- `update quota_usages set in_use='-1' where project_id='<my project id>;'`

Βιβλιογραφία - Αναφορές

- A.Souhalakis. (2010). Cloud computing και ασφάλεια.
- Amazon IaaS, PaaS, SaaS. (n.d.). Ανάκτηση από <https://aws.amazon.com/types-of-cloud-computing/>
- cloudoye. (n.d.). Ανάκτηση από <https://www.cloudoye.com/kb/general/what-is-hypervisor-in-cloud-computing-and-its-types>
- guide, O. I. (n.d.). Ανάκτηση από <https://docs.openstack.org/newton/install-guide-ubuntu/>
- Kyle, M. (2012, November 14). *https://insights.ubuntu.com*. Ανάκτηση από Virtual, Strategy magazine: <https://insights.ubuntu.com/2012/11/14/analysis-cloud-computing-private-public-or-both/>
- Microsoft, A. (n.d.). Ανάκτηση από <https://azure.microsoft.com/en-in/overview/what-is-hybrid-cloud-computing/>
- Olanubi, S. (2012, 10 12). Tharawat magazine. *Top 5 Largest Cloud Companies in the World*.
- Openstack. (n.d.). Ανάκτηση από <https://www.openstack.org/community/>
- OpenStack. (2017, 09 06). *https://docs.openstack.org*. Ανάκτηση από <https://docs.openstack.org/newton/install-guide-ubuntu/overview.html>
- Openstack foundation. (n.d.). Ανάκτηση από <https://www.openstack.org/foundation/>
- OpenStack, N. (n.d.). Ανάκτηση από <https://docs.openstack.org/newton/>
- Red Hat, Inc. (n.d.). Ανάκτηση από <https://opensource.com>
- Salesforce. (n.d.). Ανάκτηση από <https://www.salesforce.com/uk/blog/2015/11/why-move-to-the-cloud-10-benefits-of-cloud-computing.html>
- VMware. (n.d.). Ανάκτηση από <http://www.vmware.com>
- Wikipedia. (n.d.). Ανάκτηση από https://en.wikipedia.org/wiki/Cloud_computing
- Winkler, V. (. (2011). Securing the Cloud.
- Yunchuan Sun, J. Z. (2014). Data Security and Privacy in Cloud Computing.
- ανάπτυξης, P. (n.d.). Ανάκτηση από http://www.fsn.co.uk/channel_outsourcing/the_economy_is_flat_so_why_are_financials_cloud_vendors_growing_at_more_than_90_percent_per_annum#.UbmtsPIJPGA/