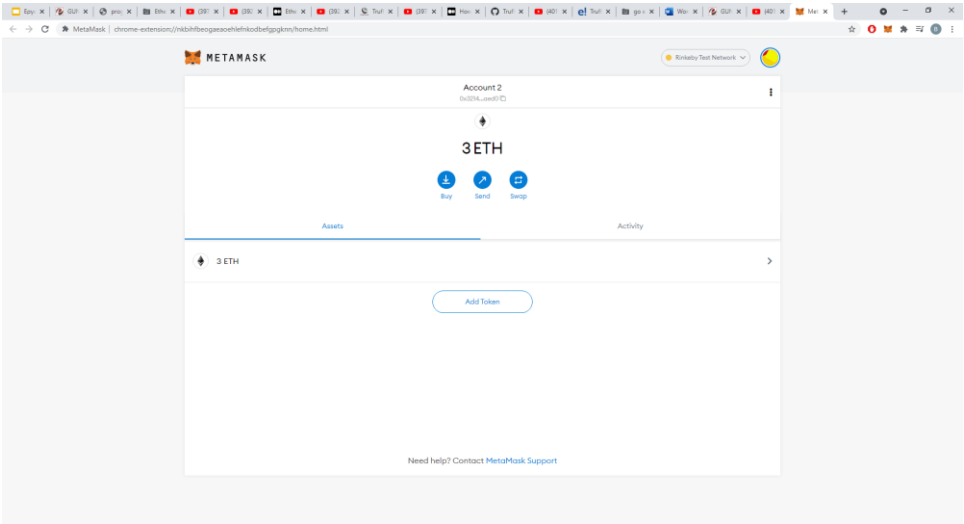
ΤΕΧΝΟΛΟΓΙΕΣ BLOCKCHAIN ΚΑΙ ΕΦΑΡΜΟΓΕΣ

Αρχικα σαν πρωτο βημα για την αναπτυξη ενος εξυπνου συμβολαιου ειναι να αποκτήσουμε ένα ψηφιακό πορτοφόλι ,το Metamask

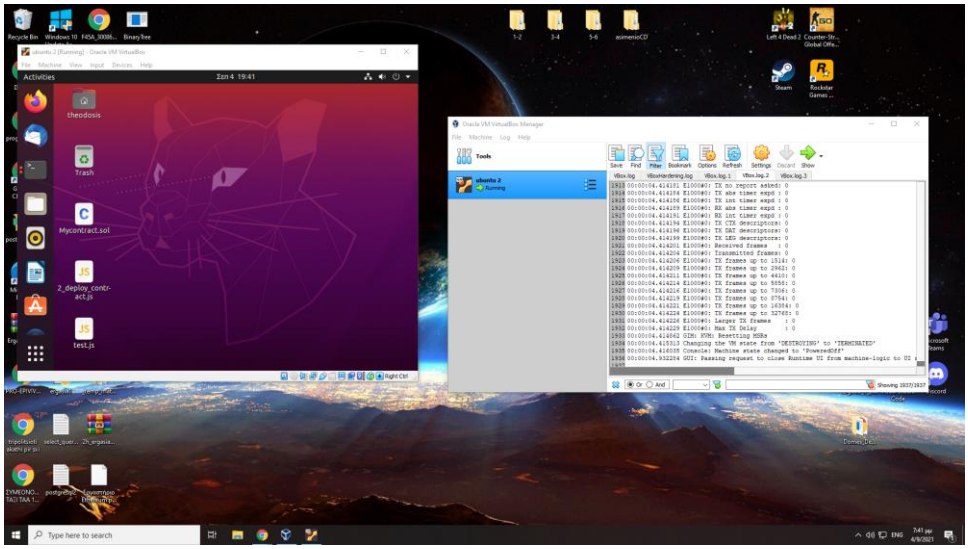
https://github.com/TheodosisSym/BlockchainTesting.git



Οπως βλεπουμε στην εικονα ειναι ενας λογαριασμος (portofolio) στο metamask οπου εχω λαβει 3 ETH απο το test faucet στο Rinkerby Test Network. Ωστε οι συναλαγες που θα κανω να μην εχουν καποιο πραγματικο κοστος

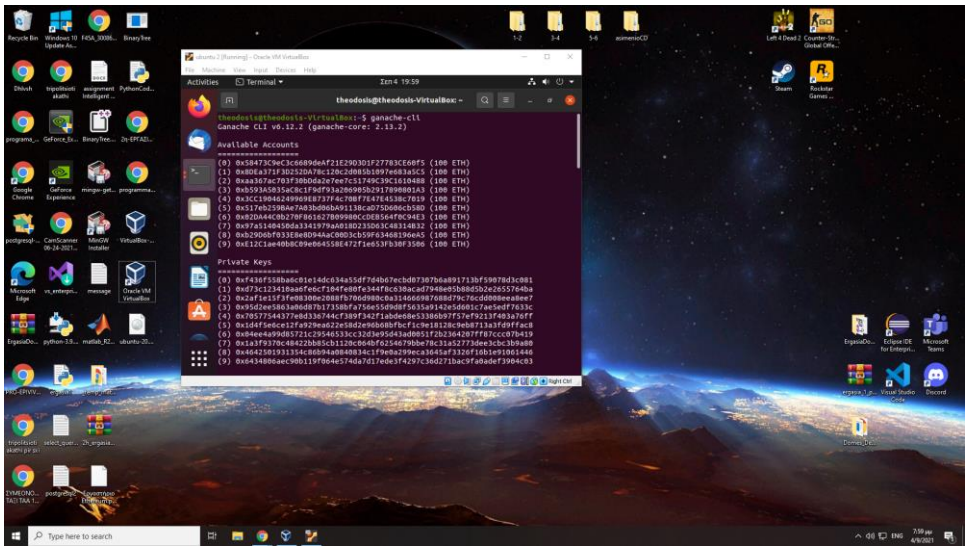
Για να τεσταρουμε ενα smart contract και να δουμε αν κανει compile ο πιο ασφαλης τροπος ειναι να το τρεξουμε σε ενα Lockal Ethereum Blockchain . Τα βηματα ειναι

1) Κατεβαζουμε μια εκδοση των Ubuntu και τα συνδεουμε στο Virtual Box



2) Τα τρεχουμε ,ανοιγουμε το Terminal και εγκαθιστουμε το nodejs και το npmjs με τις εντολες sudo apt-get install nodejs και sudo apt-get install npm

3) Εγκαθιστουμε το ganache-cli με την εντολη npm install –g ganache-cli .Το οποίο θα μας επιτρέψει να «τρέξουμε» το προσωπικό μας blockchain για testing .Το τρεχουμε και το αφηνουμε ανοιχτο αυτο το παραθυρο στο terminal

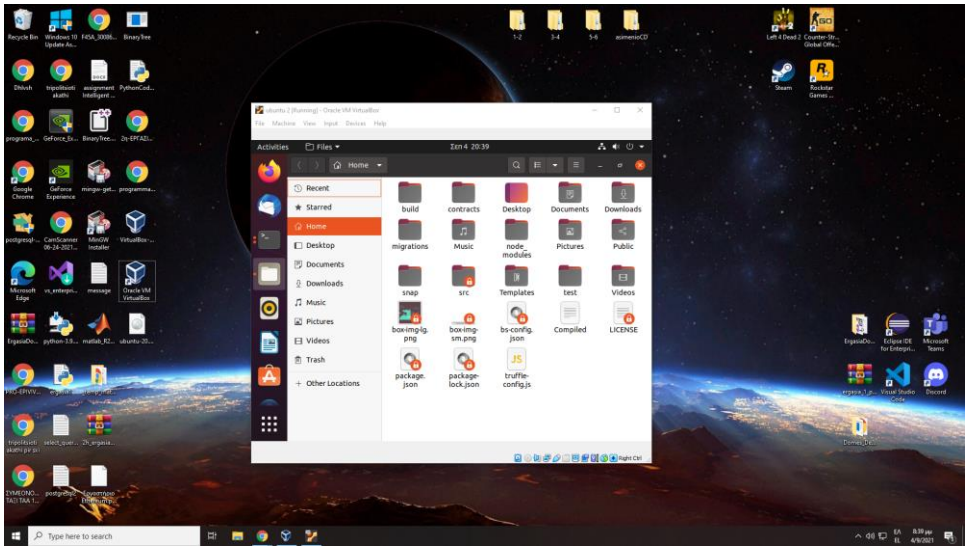


4) Ανοιγουμε ενα καινουριο παραθυρο στο terminal και εγκαθιστουμε το truffle που ειναι ενα frame work για αναπτυξη solidity με την εντολη sudo npm install –g truffle

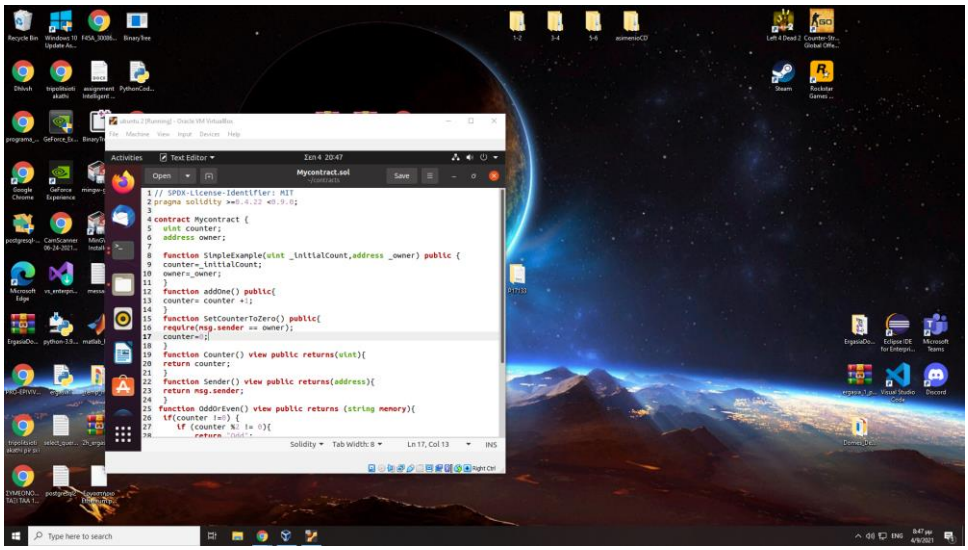
5) Αντιγράφουμε ένα secret key από τους λογαριασμούς που δημιούργησε το ganache-cli και το εισάγουμε στο metamask kai συνδεομαστε στο lockal network

8545(αυτος ειναι ο λογαριασμος που θα χρειαστουμε τα ETH απο το faucet για testing)

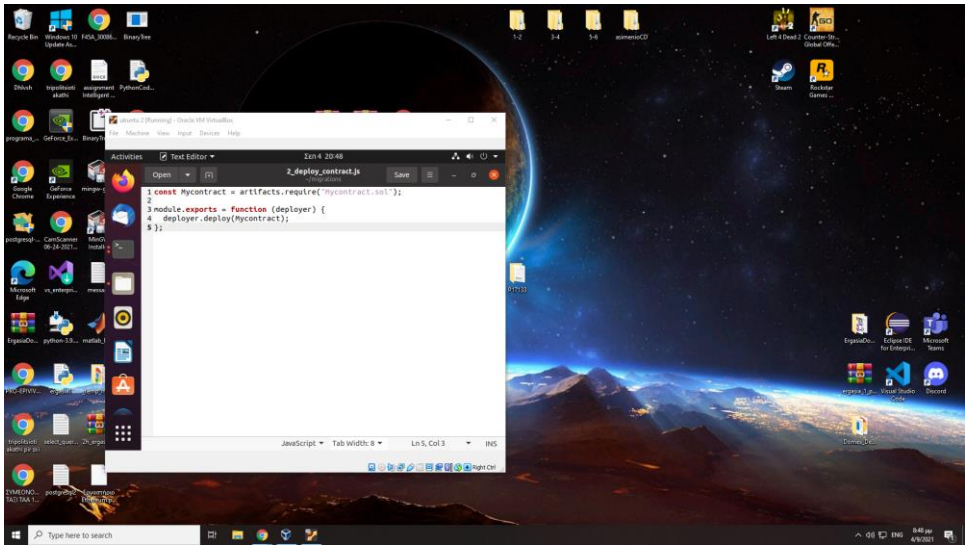
6) Σε ένα διαφορετικό terminal δημιουργούμε ένα directory. Ανοίγουμε το terminal σε αυτό το directory και πληκτρολογούμε: truffle unbox pet-shop και μας φτιαχνει ενα project με τα απαρετητα αρχεια



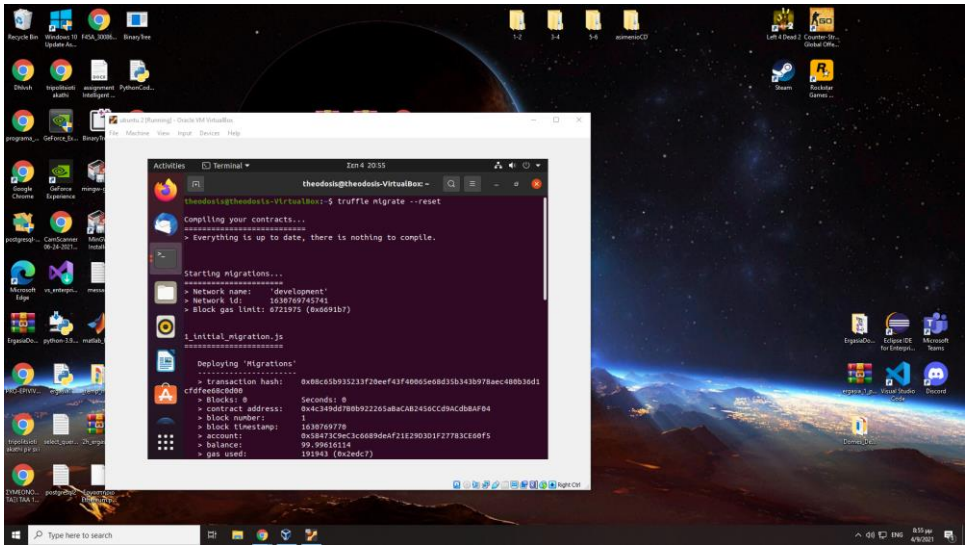
7) Δημιουργουμε ενα νεο αρχειο στο directory:/contracts που θα εινα το εξυπνο συμβολαιο με τα (5) function που εχουμε γραψει σε γλωσσα solidity και το ονομαζουμε Mycontract.sol

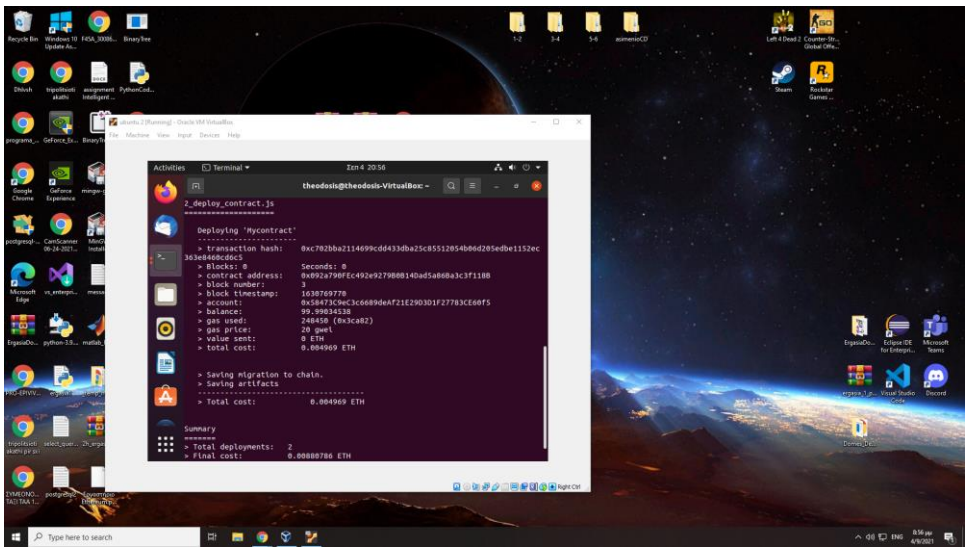


Επειτα κανουμε το ιδιο πραγμα με διαφορετικο αρχειο στο directory:/migrations και το ονομαζουμε 2_deploycontract.js

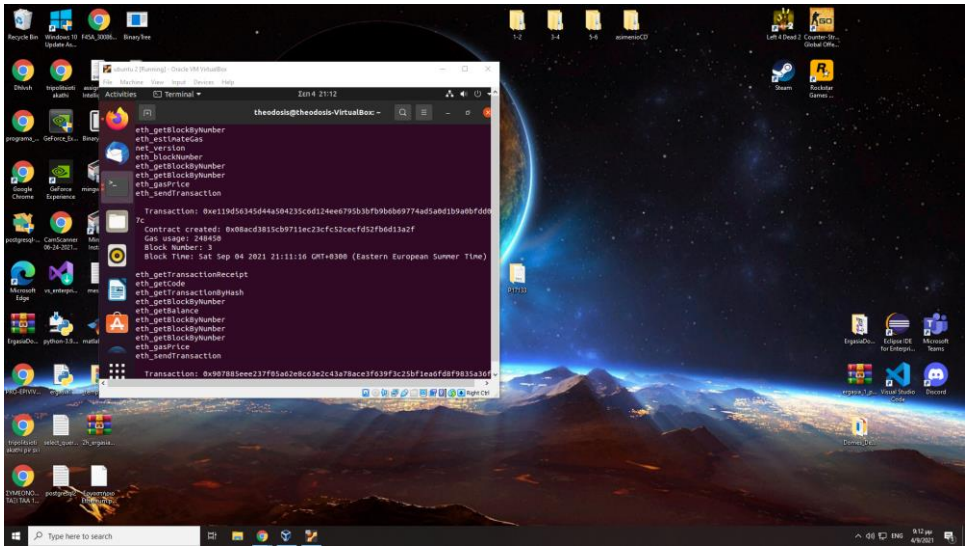


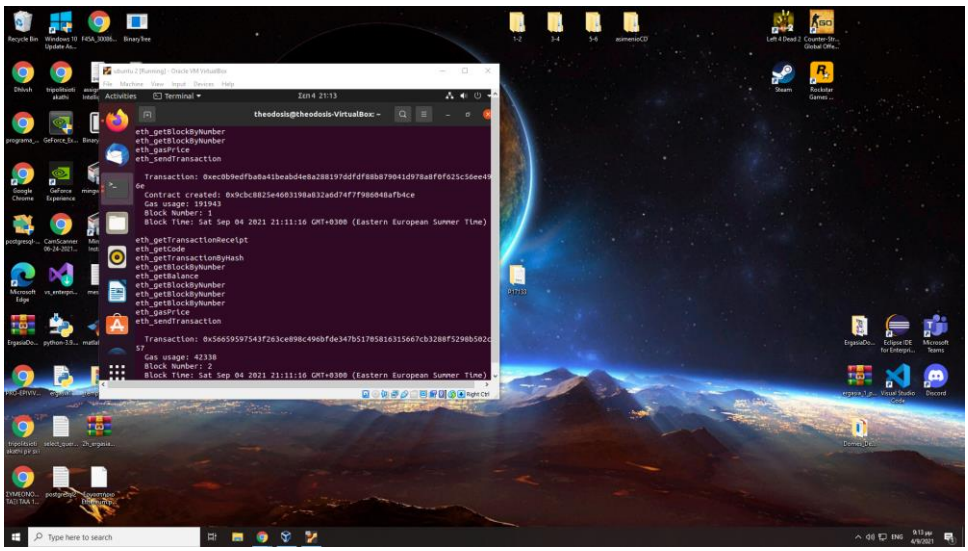
Και επειτα το τρχουμε με το truffle migrate –reset



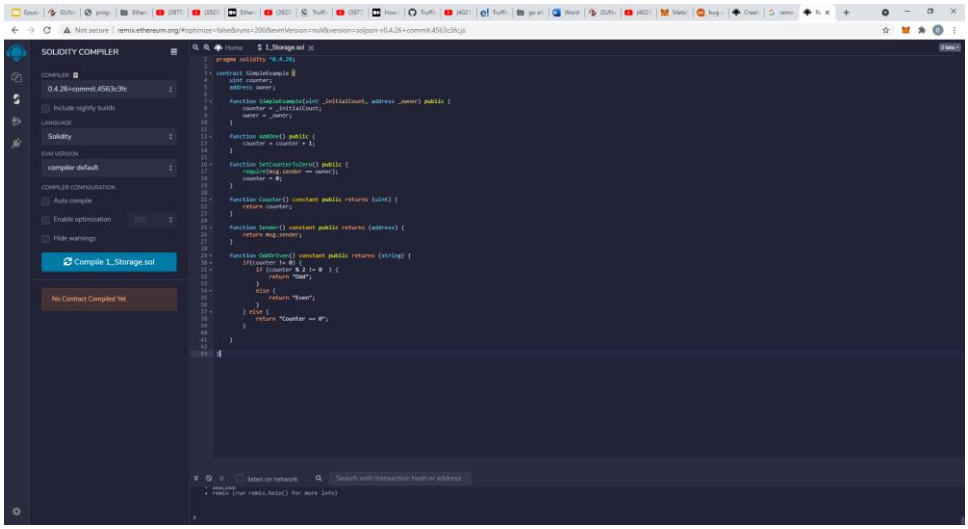


Και στα επομενα screenshot βλεπουμε απο το ganache-cli που ‘ακουει ‘ το port 8545



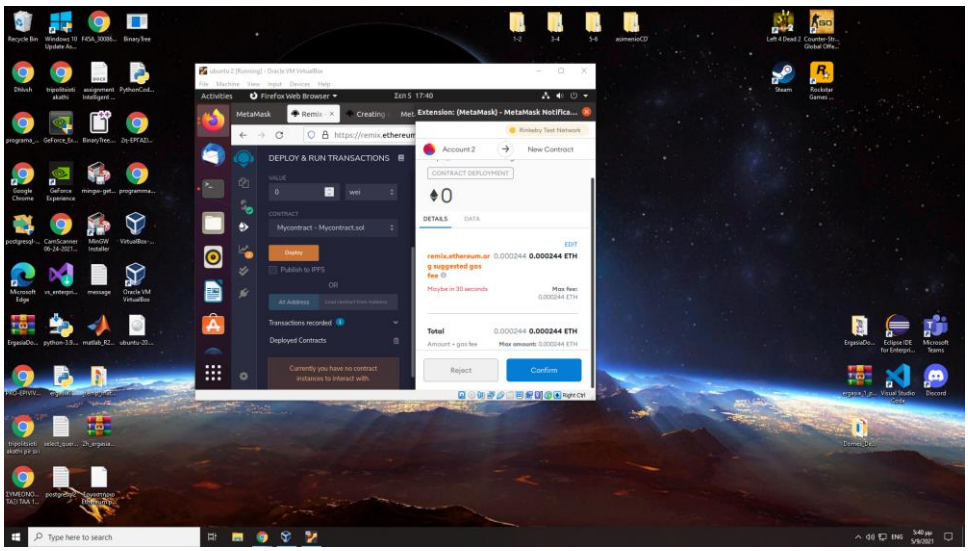


Ετσι τελος για να τρεξεις ενα εξυπνο συμβολαιο στο remix πρεπει το smart contract που θα γραψω να γινει compiled στο remix πατοντας το μπλε κουκμπι που λεει compile και το ονομα του αρχειου

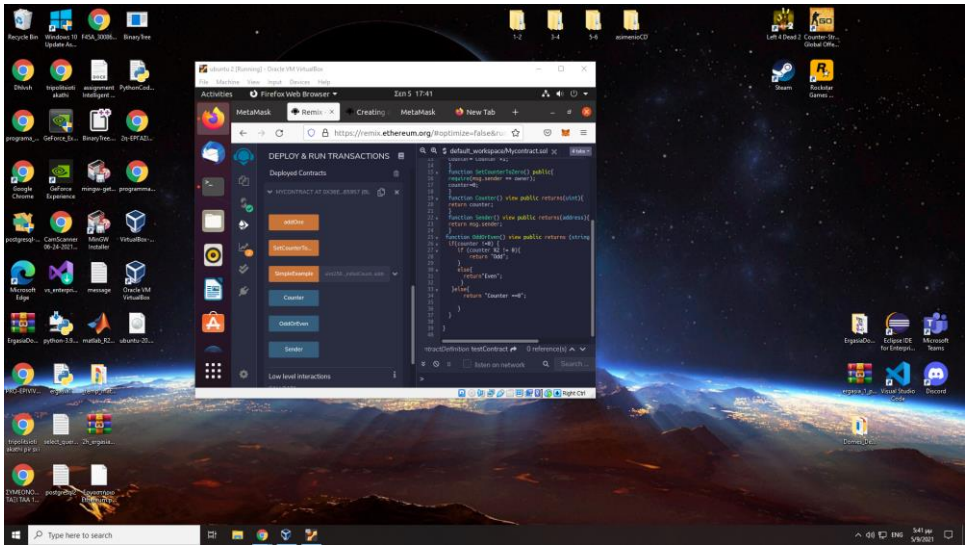


Μετα επιλεγεις το enviroment που θες να το τρεξεις και το account που θα χρησιμοποιησεις .Εγω επιλεγω το injected Web3 περιβαλλον και αμεσως μου ζηταει να συνδεσω το Metamask account μου .Τον λογαριασμο που θα συνδεσω τον εχω σε ενα Test Network για να μην εχω καποιο πραγματικο κοστος οταν θα τρεξω το συμβολαιο

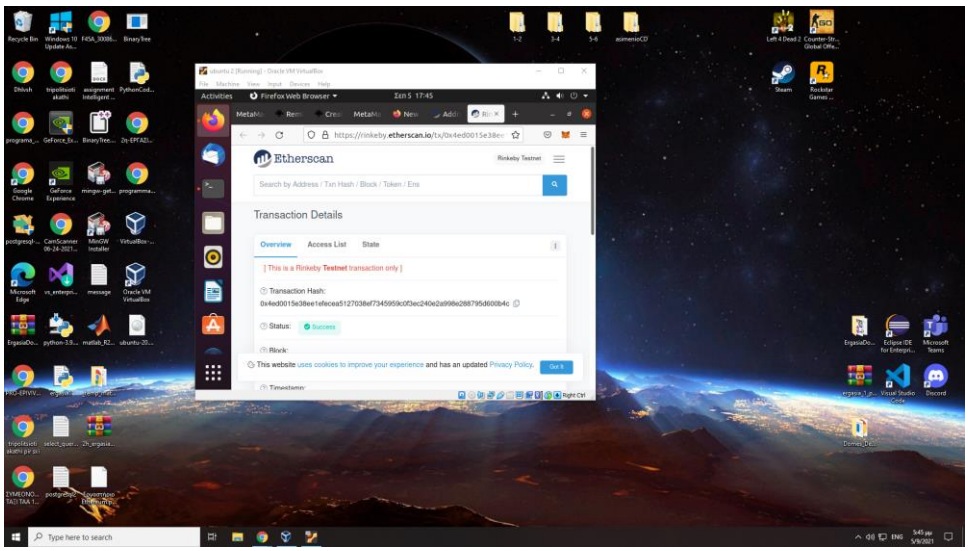
Παταμε Run για να το τρεξουμε .Κανουμε confirm την ‘πληρωμη ‘ του συμβολαιου απο το Metamask

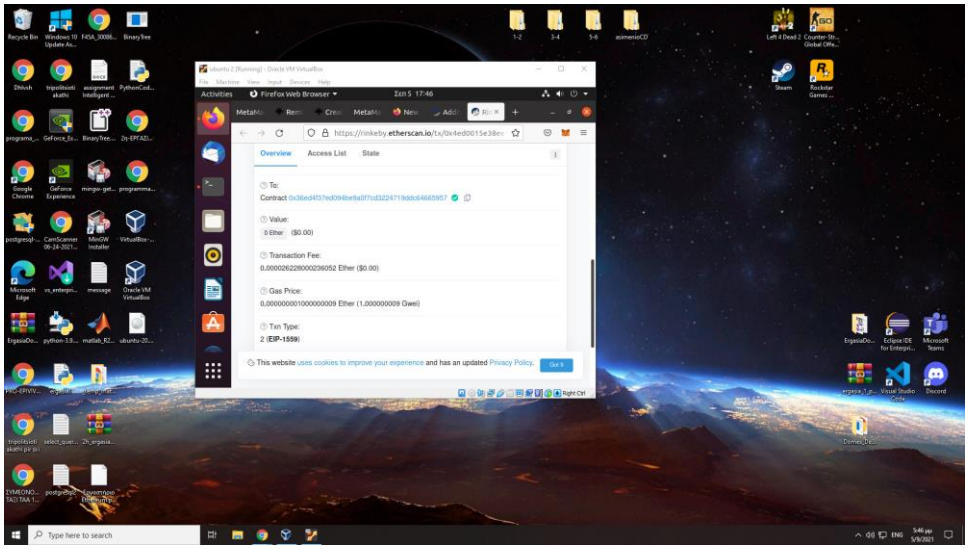


Μετα επιλεγουμε ποιο Function θα τρεξουμε και παταμε παλι confirm απο το Metamask εγω ετρεξα το



Και στην εικονα πιο κατω βλεπουμε απο το Etherscan το transaction





Πιο αναλυτικα τα Function απο το smart contract

1) Constructor

Ο constructor μας λαμβάνει δύο παραμέτρους, η πρώτη είναι να ορίσει ποιος είναι ο αρχικός μετρητής που θέλουμε να ξεκινήσει η σύμβαση και ο δεύτερος να ορίσει ποια διεύθυνση θα είναι ο «κάτοχος» αυτής της σύμβασης και να έχει προνόμια όταν καλεί τις συναρτήσεις που έχουμε δημιουργήσει.

2)AddOne

Λειτουργία AddOne που αλλάζει πληροφορίες στη σύμβαση προσθέτοντας "1" στον μετρητή.

3)SetCounterToZero
Η κλήση αυτής της λειτουργίας θα θέσει τον μετρητή μας στο μηδέν, όπου μόνο ο ιδιοκτήτης (που είχε οριστεί προηγουμένως κατά τη δημιουργία της σύμβασης) μπορεί να το κάνει.

4)Sender

Λειτουργία του sender που επιστρέφει τη διεύθυνση του αποστολέα

5)Counter

Επιστρεφει το counter

6)OddOrEven

Function που θα μας πει αν ο μετρητής είναι μονός ή ζυγός αριθμος