

# Blockchain & Αποκεντρωμένες Εφαρμογές

---

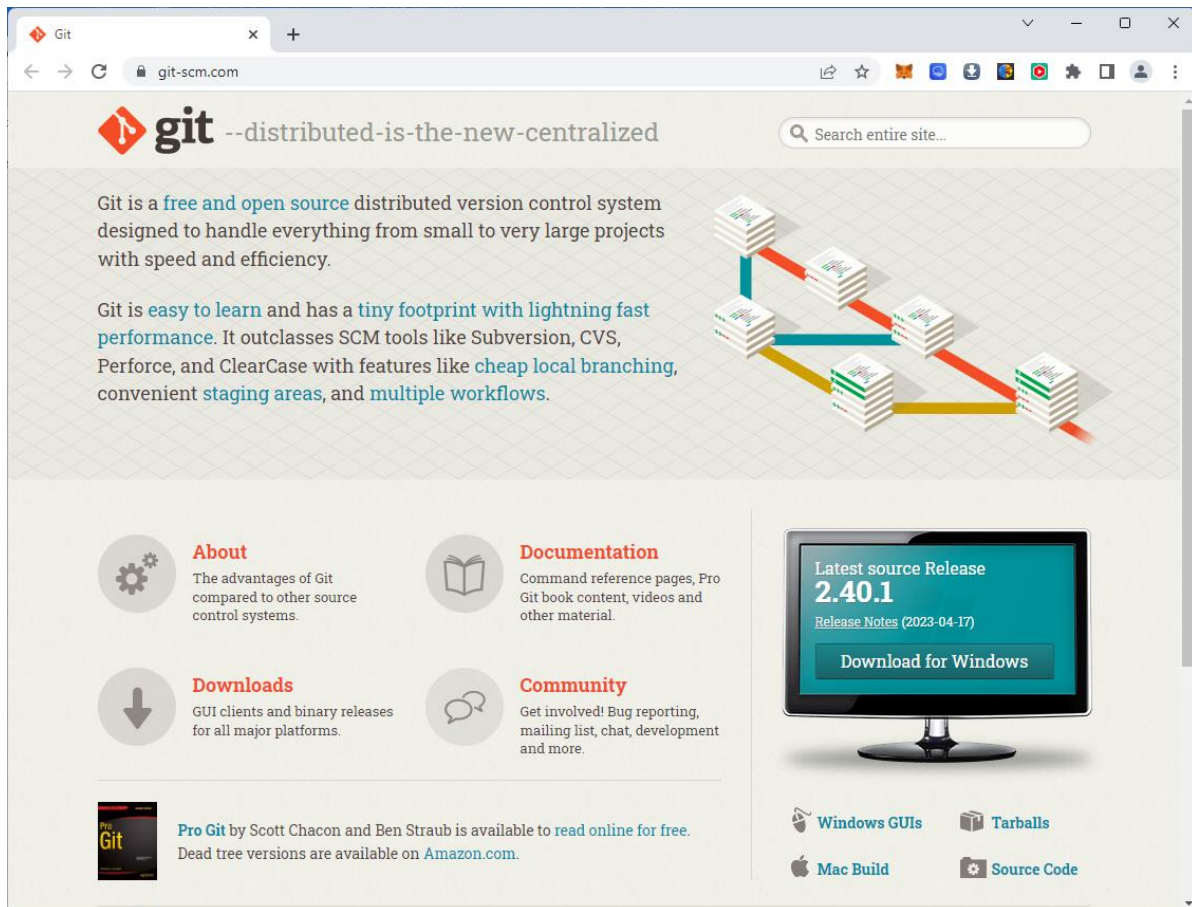
5η Εργαστηριακή Δραστηριότητα

Μαυρίδης Ιωάννης, Φουληράς Παναγιώτης  
Μάστορας Θεόδωρος  
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ



## A. Δημοσίευση στο github της ιστοσελίδας (του φακέλου) react.

1. Αν δεν το έχετε ήδη κάνει, εγκαταστήστε το **Git** από τη σελίδα <https://git-scm.com/>.



2. Ρυθμίστε το GIT (εφάπαξ) για τον τρέχοντα υπολογιστή, δίνοντας στο τερματικό τις παρακάτω εντολές, (βάζοντας τα δικά σας στοιχεία):

```
git config --global user.name "mastoras"
```

```
git config --global user.email mastoras@uom.edu.gr (προαιρετικά)
```

```
git config --global user.password "1234321"
```

3. Δημιουργήστε στο GitHub ένα νέο repository (π.χ. "smart09"). Εντός του φακέλου react (του site που θέλετε να δημοσιεύσετε), δώστε τις παρακάτω εντολές στο τερματικό, φροντίζοντας να δώσετε τα δικά σας στοιχεία (εδώ χρησιμοποιούμε το branch main αντί του master, επειδή στο master θα δημοσιευτεί το static site).

```
git init
```

```
git branch -M main
```

```
git remote add origin https://github.com/mastoras/smart09.git
```

```
<αν υπάρχει το repo αλλά όχι το branch> git push -u origin main
```

```
<αν υπάρχει> git pull https://github.com/mastoras/smart09.git main
```

```
echo "# smart09" >> README.md
```

```
git add README.md
```

```
git commit -m "first commit"
```

4. Στο Visual Studio Code ανοίξετε το **φάκελο του site**. Κάνετε (με τη βοήθεια του πρόσθετου “GitHub Pull Requests and Issues”) Commit και Publish. Αν είναι η πρώτη φορά που χρησιμοποιείτε το VS Code για να κάνετε Publish, θα σας ζητηθεί να κάνετε είσοδο από την ιστοσελίδα του Github και να δώσετε την έγκρισή σας (στο VS Code). Πηγαίνετε μετά το Publish στην ιστοσελίδα του repository και ελέγξτε πως όλα τα αρχεία «ανέβηκαν». Παρατηρήστε πως δημιουργήθηκε/χρησιμοποιήθηκε το branch “main”.

5. Εντός του φακέλου react, δώστε την παρακάτω εντολή για να εγκαταστήσετε τη βιβλιοθήκη node gh-pages. Η επέκταση αυτή του node επιτελεί διπλό ρόλο, από τη μία «**συμπιέζει**» **ολόκληρο το site σε στατική μορφή** εντός πολύ λίγων αρχείων, και από την άλλη **αναρτά στο github** αυτά τα αρχεία ώστε να είναι έτοιμα για δημοσίευση.

```
npm install gh-pages --save-dev
```

6. Στο package.json του φακέλου react κάνετε τις παρακάτω προσθήκες (πορτοκαλί φόντο – αλλάζτε δίνοντας τα δικά σας στοιχεία). Παρατηρήστε πως ορίζεται το deploy να γίνει στο branch **master** ώστε να μην υπάρξει καμία επικάλυψη με ό,τι ενδεχομένως έχετε ήδη στο **main**:

```
{  
  "name": "smart09",  
  "version": "0.1.0",  
  "homepage": "https://mastoras.github.io/smart09",  
  "private": true,  
  "dependencies": {  
    "@testing-library/jest-dom": "^5.17.0",  
    "@testing-library/react": "^13.4.0",  
    "@testing-library/user-event": "^13.5.0",  
    "bootstrap": "^5.3.3",  
    "react": "^18.3.1",  
    "react-dom": "^18.3.1",  
    "react-scripts": "5.0.1",  
    "web-vitals": "^2.1.4",  
    "web3": "^4.8.0"  
  },  
  "scripts": {  
    "predeploy": "npm run build",  
    "deploy": "gh-pages -b master -d build",  
    "start": "react-scripts start",  
    "build": "react-scripts build",  
  }  
}
```

```

    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "gh-pages": "^6.1.1"
  }
}

```

7. Στο <https://github.com/settings/tokens> ζητήστε ένα νέο token.

The screenshot shows the GitHub 'Personal access tokens (classic)' settings page. On the left, there is a sidebar with navigation options: 'GitHub Apps', 'OAuth Apps', 'Personal access tokens' (selected), 'Fine-grained tokens (Beta)', and 'Tokens (classic)'. The main content area is titled 'Personal access tokens (classic)' and includes a 'Generate new token' button circled in red. A dropdown menu is open from this button, showing two options: 'Generate new token (Beta)' (Fine-grained, repo-scoped) and 'Generate new token (classic)' (For general use), with a red arrow pointing to the classic option. Below the dropdown, there is a table of existing tokens:

Token Name	Scopes	Expiration	Usage	Action
deploy-access	read:org, repo	Expired on Mon, Mar 20 2023.		Delete
Remix IDE Token	gist, repo	Expired on Fri, Feb 24 2023.	Never used	Delete

At the bottom, there is a note: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.'

Settings / Developer settings

GitHub Apps  
 OAuth Apps  
 Personal access tokens  
 Fine-grained tokens (Beta)  
 Tokens (classic)

### New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**  
 deploy-access  
 What's this token for?

**Expiration \***  
 30 days The token will expire on Tue, Jun 27 2023

**Select scopes**  
 Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

- repo** Full control of private repositories
- repo:status Access commit status
- repo\_deployment Access deployment status
- public\_repo Access public repositories
- repo:invite Access repository invitations
- security\_events Read and write security events
- workflow Update GitHub Action workflows
- write:packages Upload packages to GitHub Package Registry
- read:packages Download packages from GitHub Package Registry
- delete:packages Delete packages from GitHub Package Registry
- admin:org Full control of orgs and teams, read and write org projects
- write:org Read and write org and team membership, read and write org projects
- read:org Read org and team membership, read org projects
- manage\_runners:org Manage org runners and runner groups
- admin:public\_key Full control of user public keys

Generate token

Δώστε ένα όνομα και ρυθμίστε τα δικαιώματα του κατόχου. Πατήστε το **Αντιγράψτε** κάπου τον κωδικό που θα σας εμφανιστεί **μία και μοναδική φορά**.

Personal access tokens (classic) Generate new token Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp\_Q... Delete

8. Εντός του φακέλου `react`, δώστε την παρακάτω εντολή:

```
git remote set-url origin https://<user>:<token>@github.com/<user>/<repo>
```

π.χ.

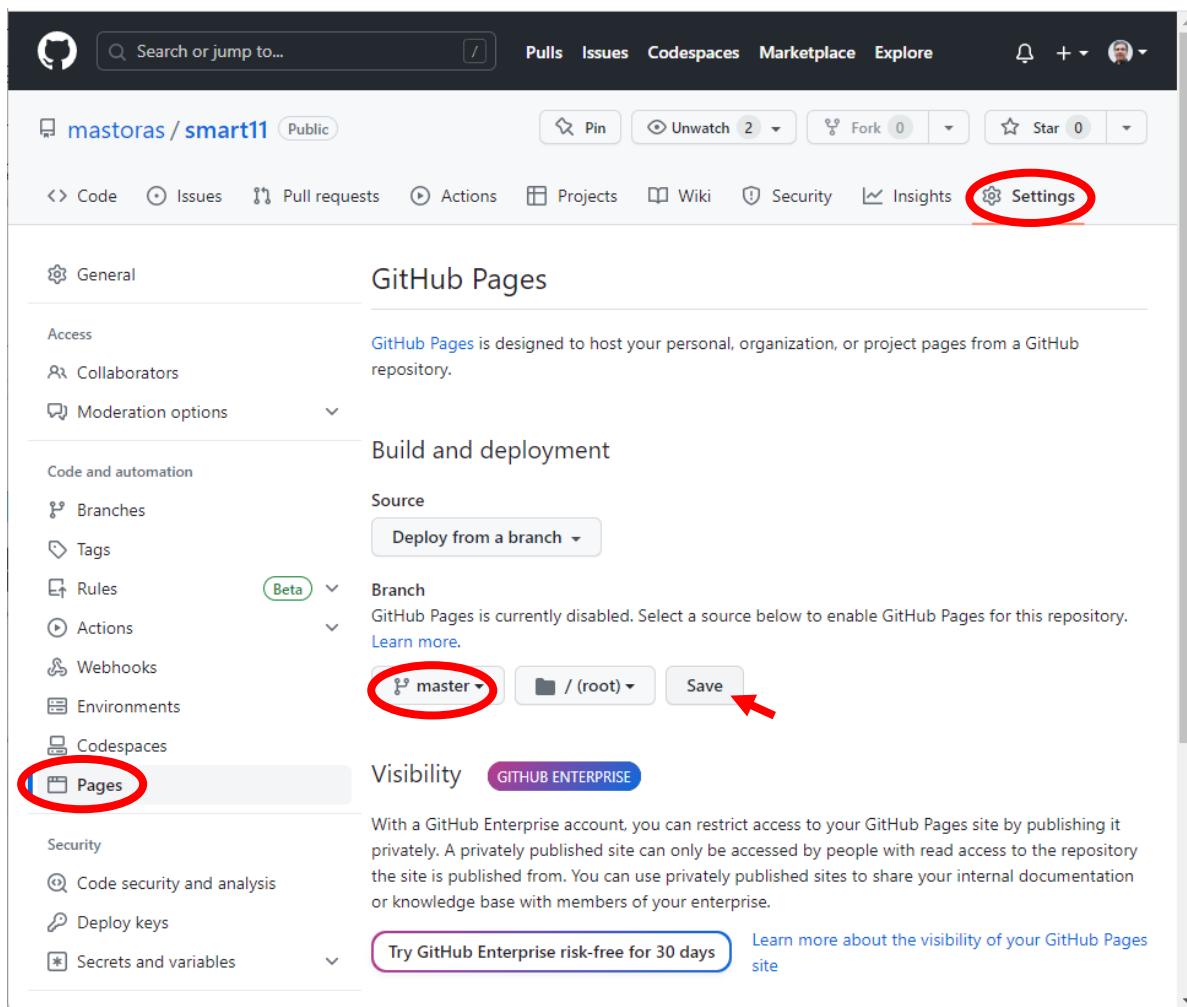
```
git remote set-url origin https://mastoras:gh..i@github.com/mastoras/smart09
```

**9.** Εντός του φακέλου `react`, δώστε την παρακάτω εντολή για να δημοσιεύσετε την στατική μορφή του site στο branch `master`:

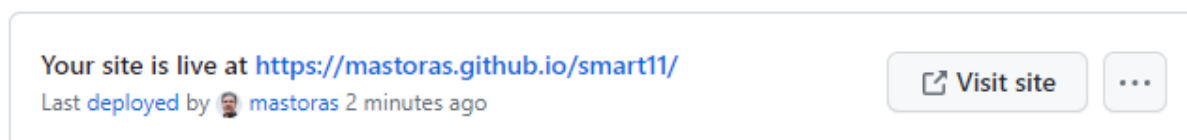
```
npm run deploy
```

Παρατηρήστε πως εκτελείται η εντολή `npm run build` η οποία δημιουργεί την στατική μορφή των αρχείων στο φάκελο `build`.

**10.** Μην ξεχάσετε να πάτε στο Github και από το Settings -> Pages να ενεργοποιήσετε τον web server για το branch `master`.



Προσοχή! Χρειάζονται περίπου 5 λεπτά μέχρι να αδειοδοτηθεί προς δημοσίευση ο φάκελος. Όταν θα δημοσιευτεί, θα εμφανιστεί ένα μήνυμα στο πάνω μέρος της σελίδας:



Η σελίδα σας βρίσκεται στο link <https://<user>.github.io/<repository>/>

**Παρατήρηση:** Υπάρχουν υπηρεσίες φιλοξενίας ενός React project οι οποίες είναι δωρεάν μέχρι κάποιο όριο traffic. Μπορείτε αν θέλετε να δοκιμάσετε τις <https://www.netlify.com/> ή <https://vercel.com/>. Προσοχή! Ζητάνε πρόσβαση στον λογαριασμό σας στο Github και μετά από υπέρβαση των ορίων διακόπτουν τη λειτουργία τους ή εφαρμόζουν χρεώσεις.

## B. Δημοσίευση τοπικά στον IIS της ιστοσελίδας (του φακέλου) react

Η στατική εκδοχή της σελίδας react που προέκυψε από την προηγούμενη build έχει ενσωματωμένο το URL πρόθεμα "<https://mastoras.github.io/smart09>" λόγω των ρυθμίσεων στο αρχείο package.json. Αυτή τη φορά θέλουμε όλες οι ιστοσελίδες να ξεκινούν από τον εκάστοτε φάκελο "home" δηλαδή τον...

./

1. Στο package.json του φακέλου react κάνετε λοιπόν ΜΟΝΟ την παρακάτω προσθήκη (πορτοκαλί φόντο):

```
{
{
  "name": "react01",
  "version": "0.1.0",
  "homepage": ".",
  "private": true,
  ...
```

2. Εντός του φακέλου react, δώστε την παρακάτω εντολή για να δημιουργήσετε την στατική μορφή του site στο φάκελο "build":

**npm run build**

3. Εγκαταστήστε τον Internet Information Server (IIS) στη VM αν δεν το έχετε ήδη κάνει από την 3<sup>η</sup> εργαστηριακή δραστηριότητα.
4. Αντιγράψτε τον φάκελο **build** στον φάκελο «C:\inetpub\wwwroot». Σύρετε τον φάκελο από παράθυρο σε παράθυρο Windows Explorer κρατώντας πατημένο το πλήκτρο Ctrl. Ο φάκελος «wwwroot» είναι ο προεπιλεγμένος ριζικός φάκελος του IIS.
5. Ρυθμίστε το firewall των windows ώστε να επιτρέπονται οι εισερχόμενες κλήσεις στο port 80. Μπορείτε πλέον να ανοίξετε την ιστοσελίδα [http:// 83.212.96.54/build/](http://83.212.96.54/build/). Δώστε την ip της δικής σας VM.



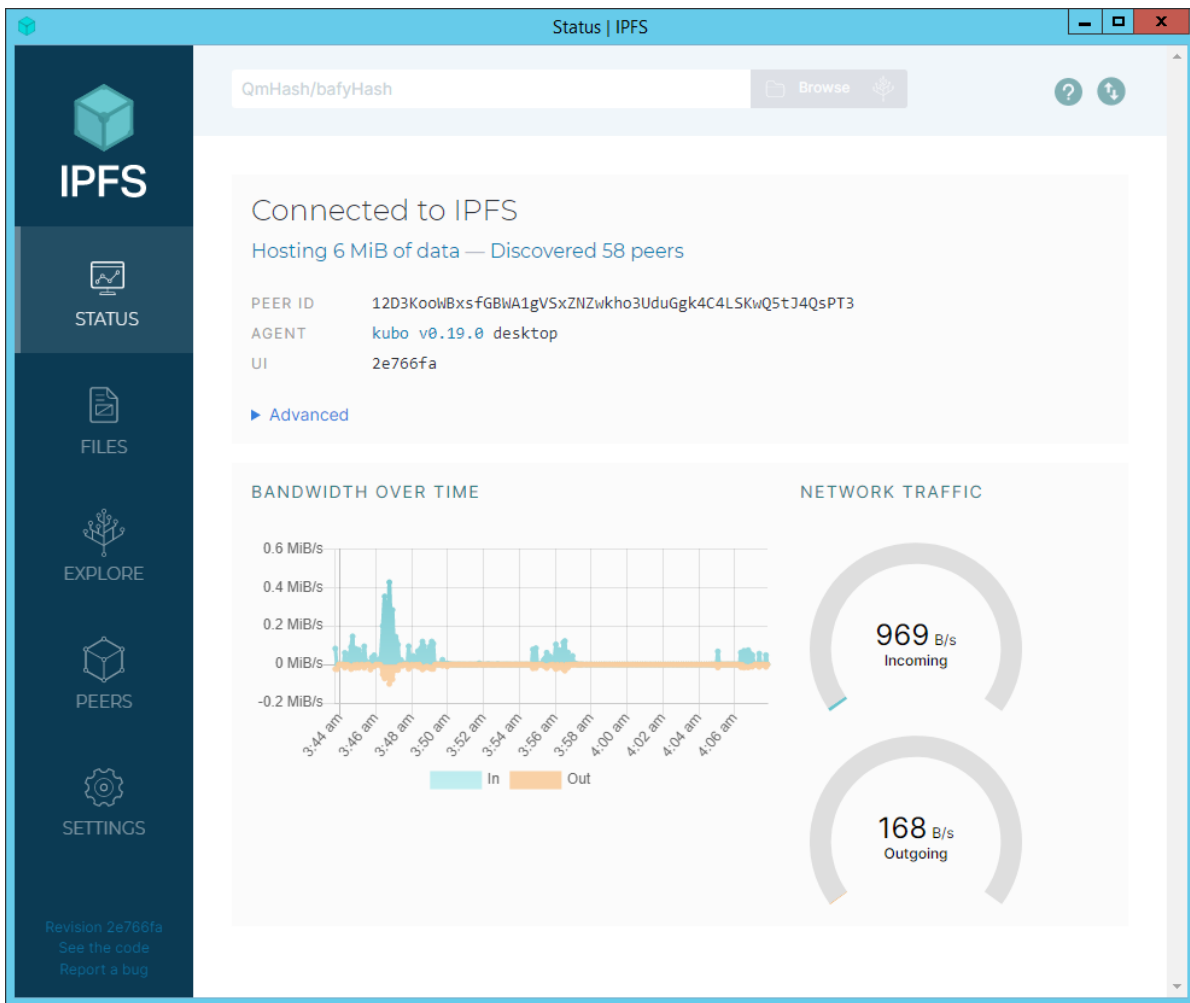
## C. Δημοσίευση στο IPFS (Πραγματικό DApp) της ιστοσελίδας react

Κατεβάστε από <https://github.com/ipfs/ipfs-desktop/releases> και εγκαταστήστε το IPFS Desktop.

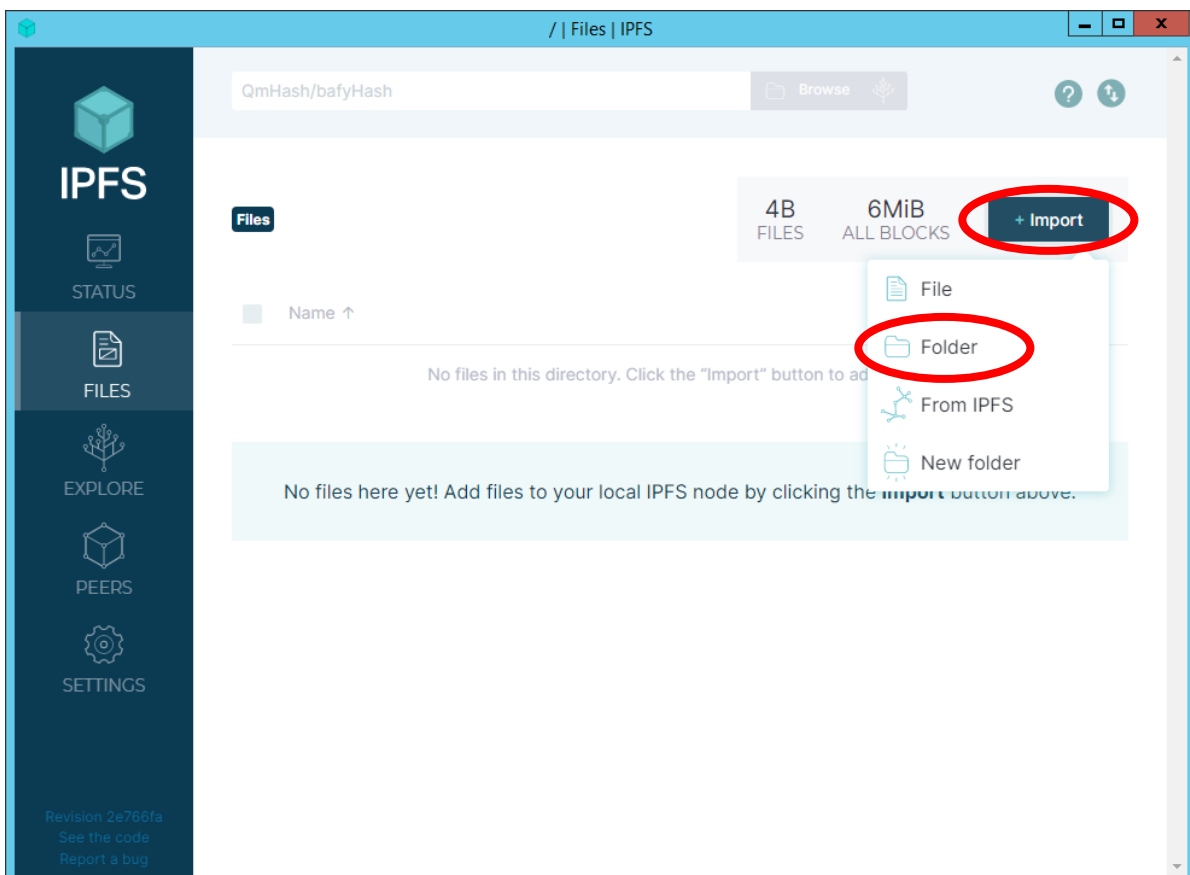
The screenshot shows the GitHub release page for `ipfs/ipfs-desktop` version `0.28.0`. The page includes a search bar, navigation tabs (Code, Issues, Pull requests, Actions, Security, Insights), and a 'Releases' tab. The release details section shows the version number, a 'Latest' badge, and a 'What's Changed' section with a bullet point: 'Update Kubo: v0.20.0 by @hacdias in #2481'. Below this is a 'Full Changelog' link. The 'Contributors' section lists 'hacdias'. The 'Assets' section shows 17 assets, including various platform-specific binaries and source code. A red circle highlights the 'Show all 17 assets' link, and a red arrow points to the expanded list of assets below, which includes 'IPFS-Desktop-Setup-0.28.0.exe'.

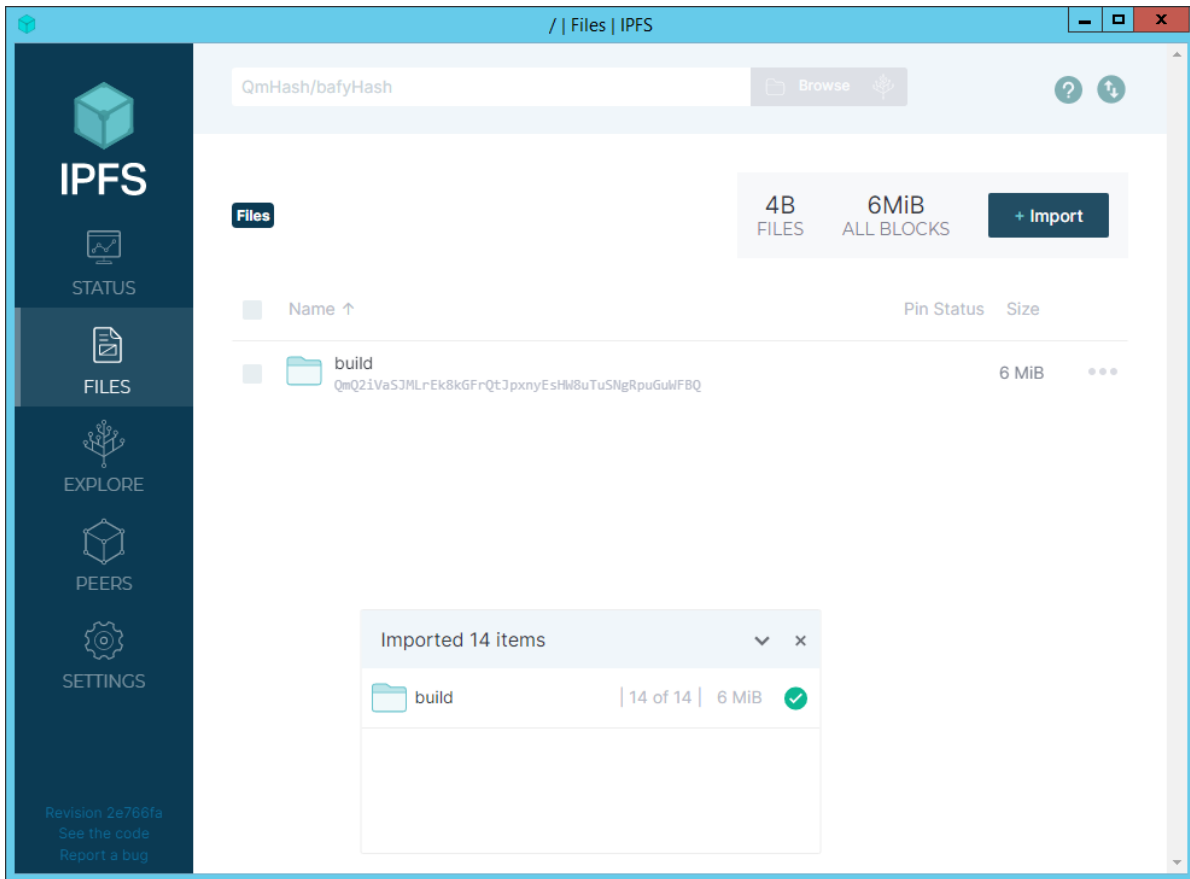
Asset Name	Size	Time
ipfs-desktop-0.28.0-linux-amd64.deb	96.4 MB	3 weeks ago
ipfs-desktop-0.28.0-linux-amd64.snap	114 MB	3 weeks ago
ipfs-desktop-0.28.0-linux-x64.freebsd	98.4 MB	3 weeks ago
ipfs-desktop-0.28.0-linux-x64.tar.xz	95.8 MB	3 weeks ago
ipfs-desktop-0.28.0-linux-x86_64.Applmage	131 MB	3 weeks ago
ipfs-desktop-0.28.0-linux-x86_64.rpm	95.9 MB	3 weeks ago
ipfs-desktop-0.28.0-mac.dmg	127 MB	3 weeks ago
ipfs-desktop-0.28.0-mac.dmg.blockmap	135 KB	3 weeks ago
ipfs-desktop-0.28.0-squirrel.zip	121 MB	3 weeks ago
ipfs-desktop-0.28.0-squirrel.zip.blockmap	130 KB	3 weeks ago
Source code (zip)		3 weeks ago
Source code (tar.gz)		3 weeks ago
Show all 17 assets		

Εκτελέστε το IPFS Desktop

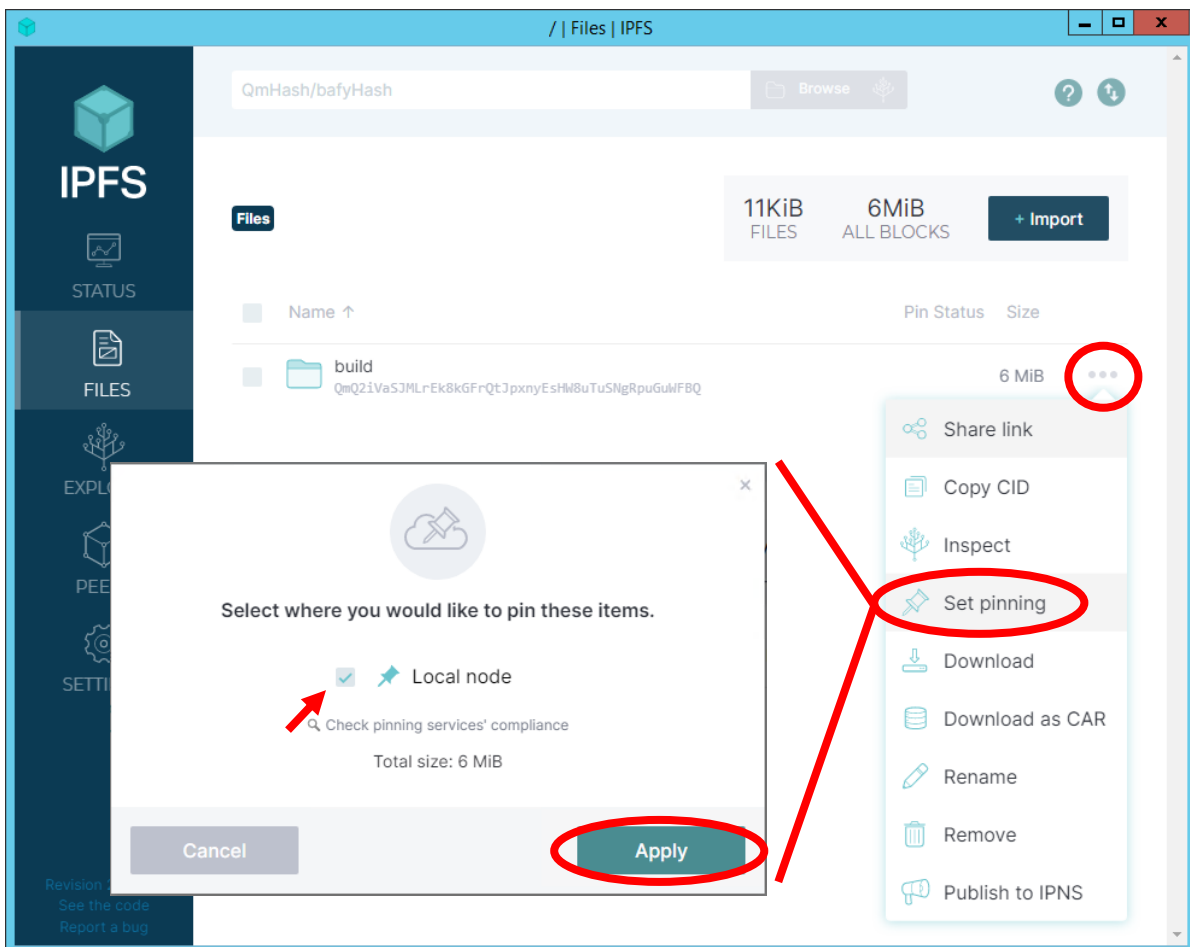


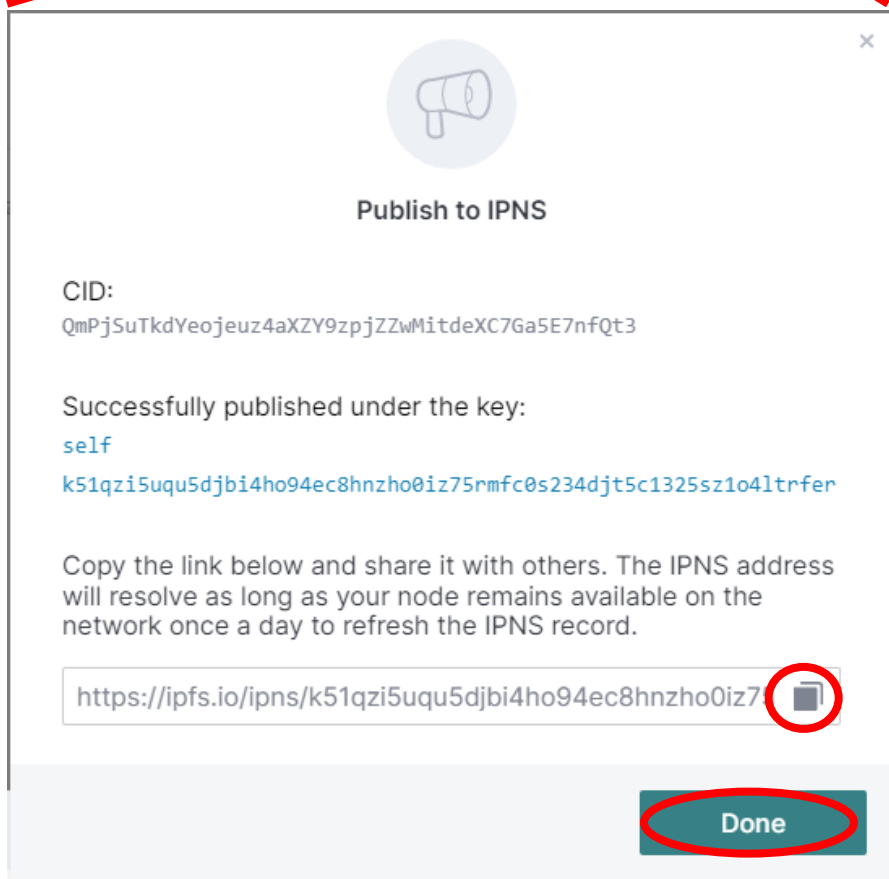
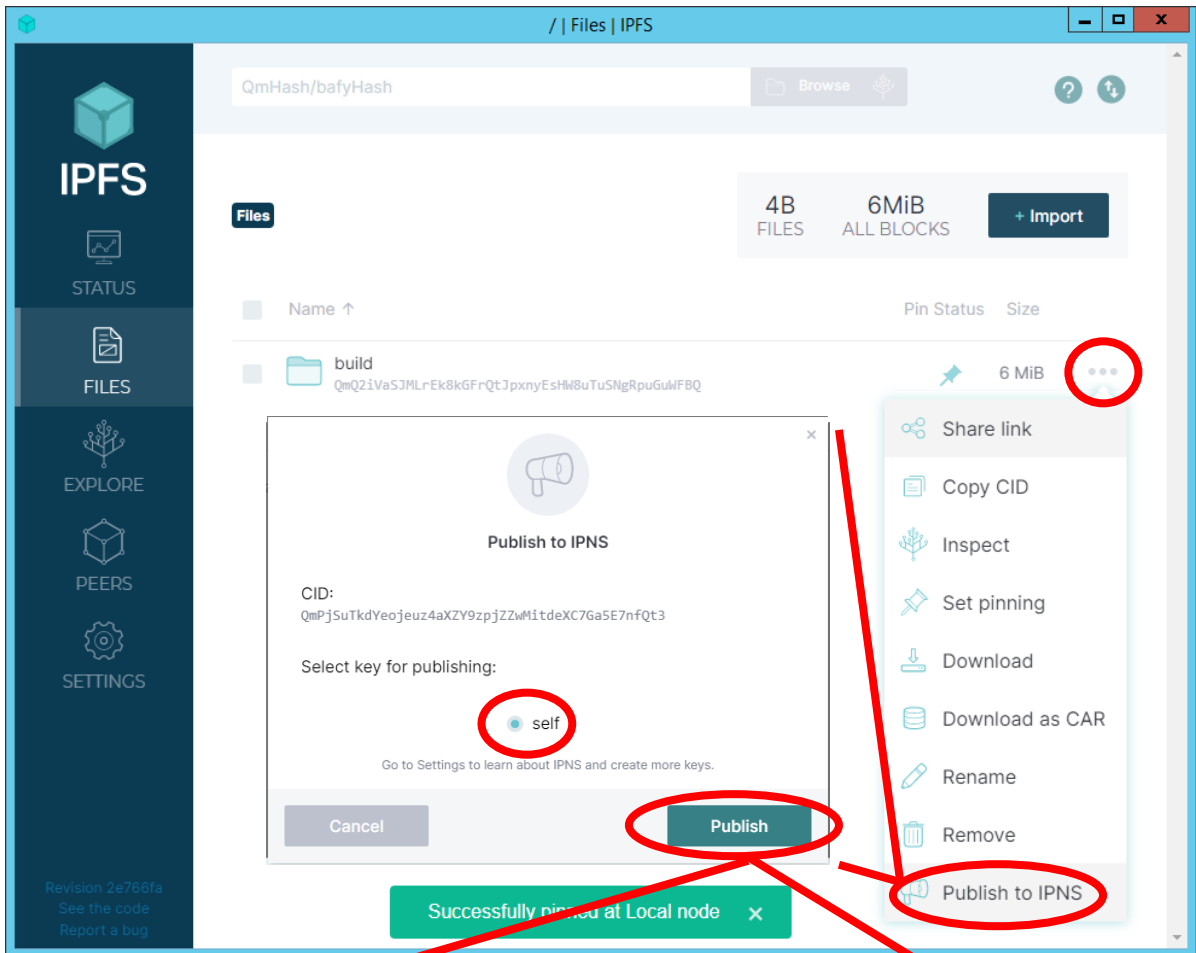
Εισάγετε τον φάκελο build που προέκυψε ως η στατική εκδοχή του react φακέλου (με το "homepage": ".", στο package.json).



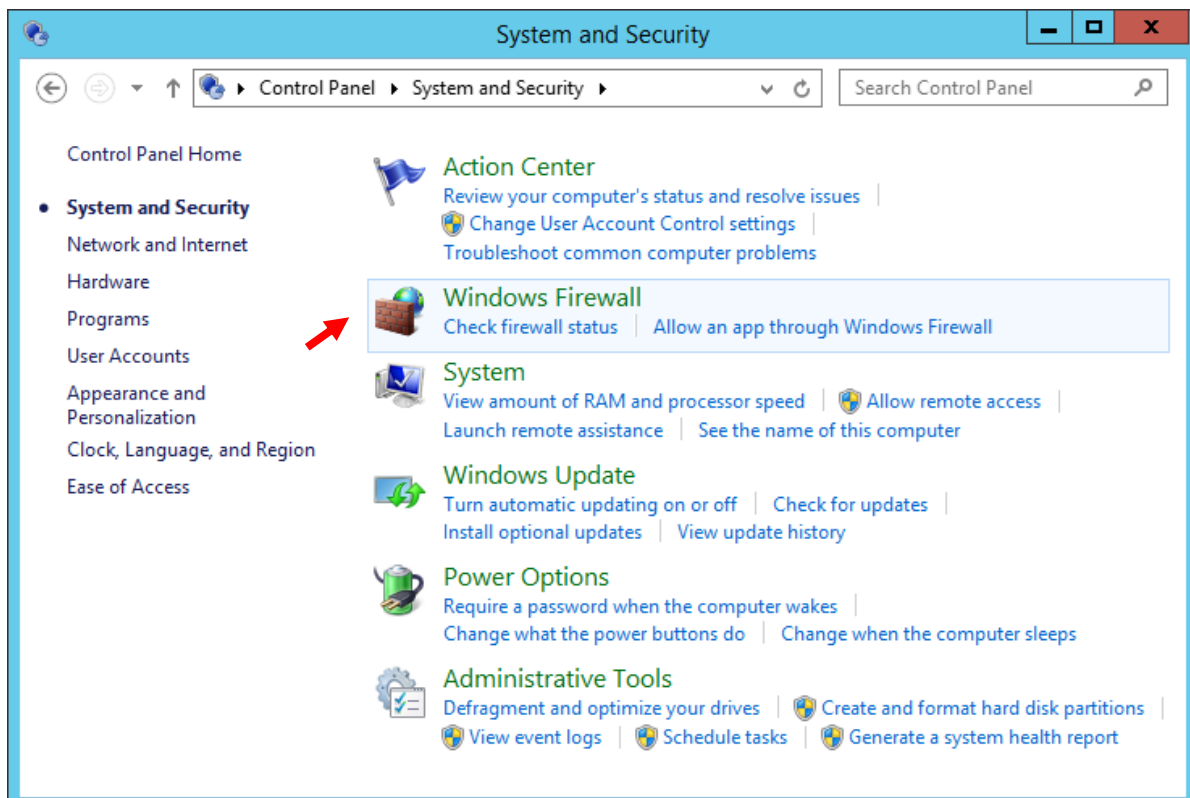
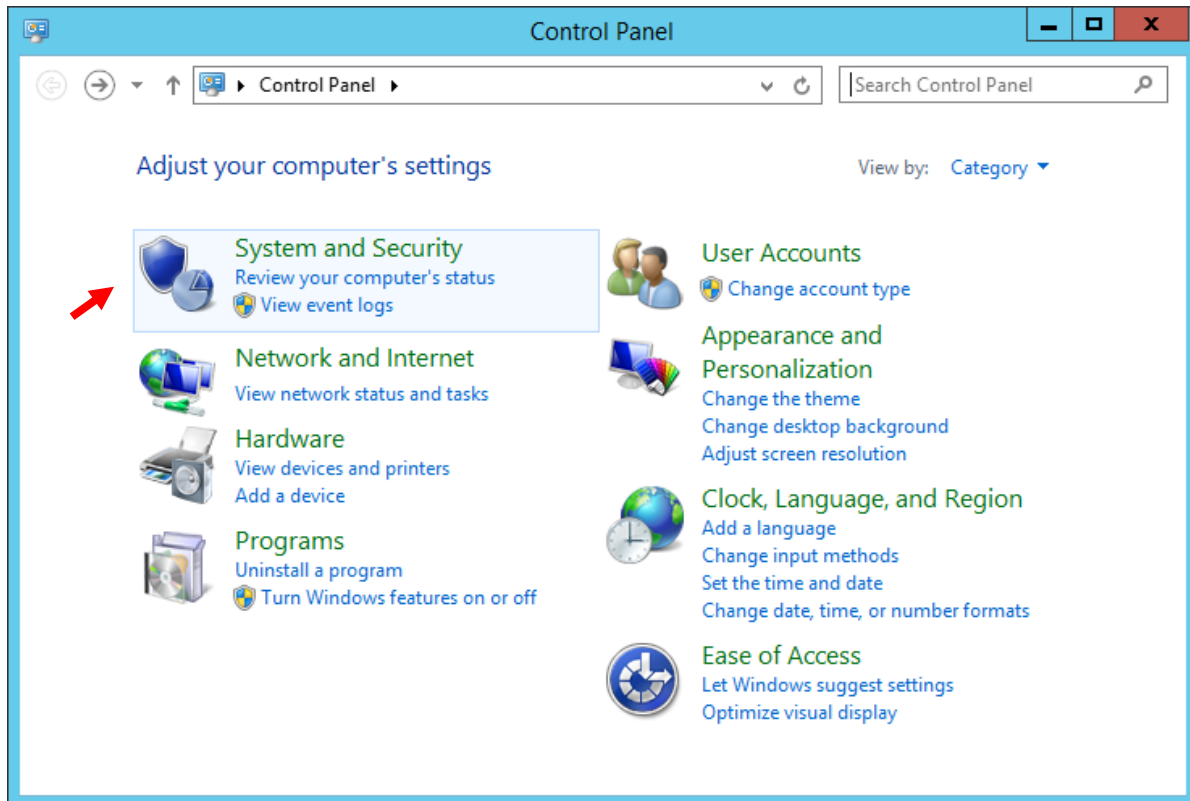


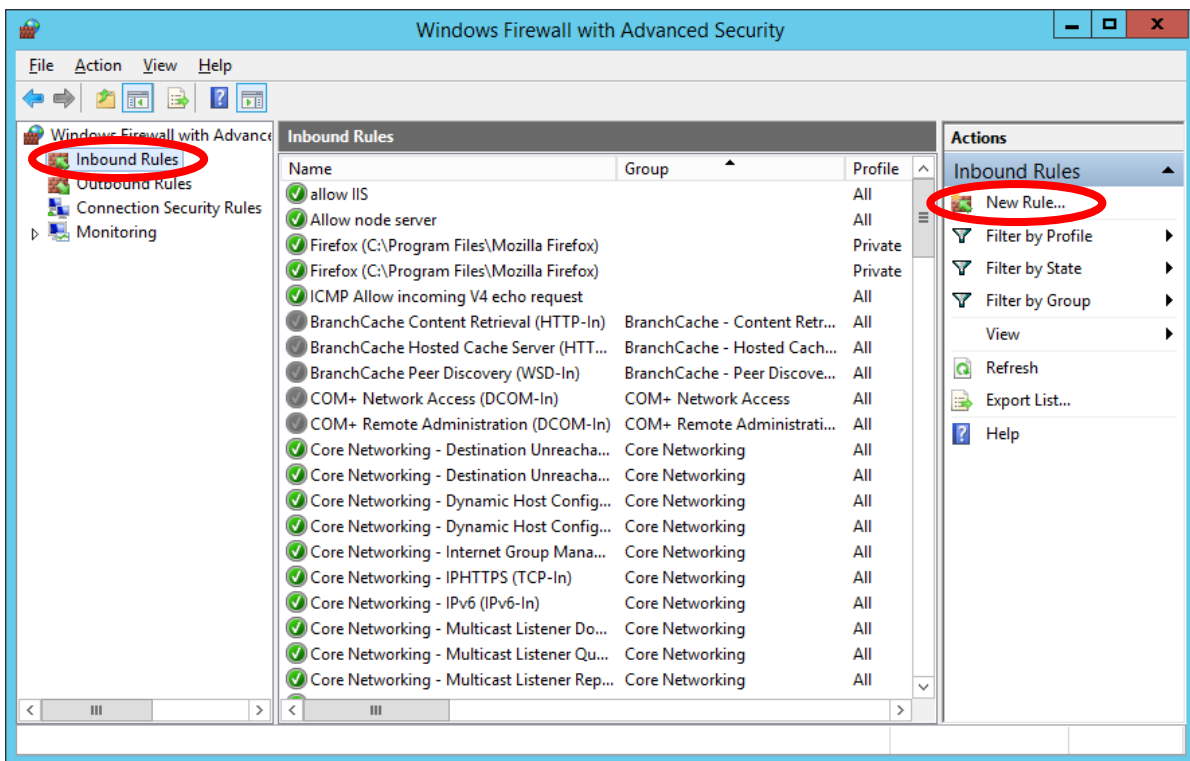
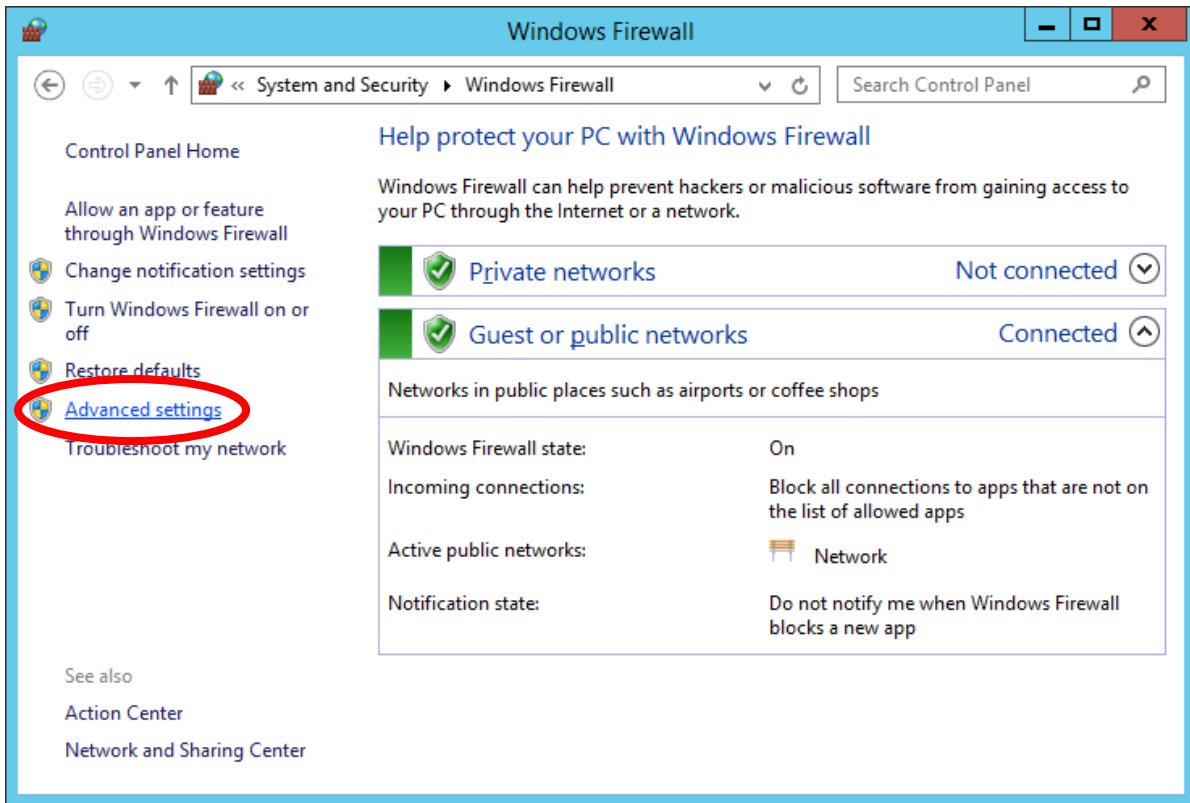
Καρφίτωστε τον τοπικά.

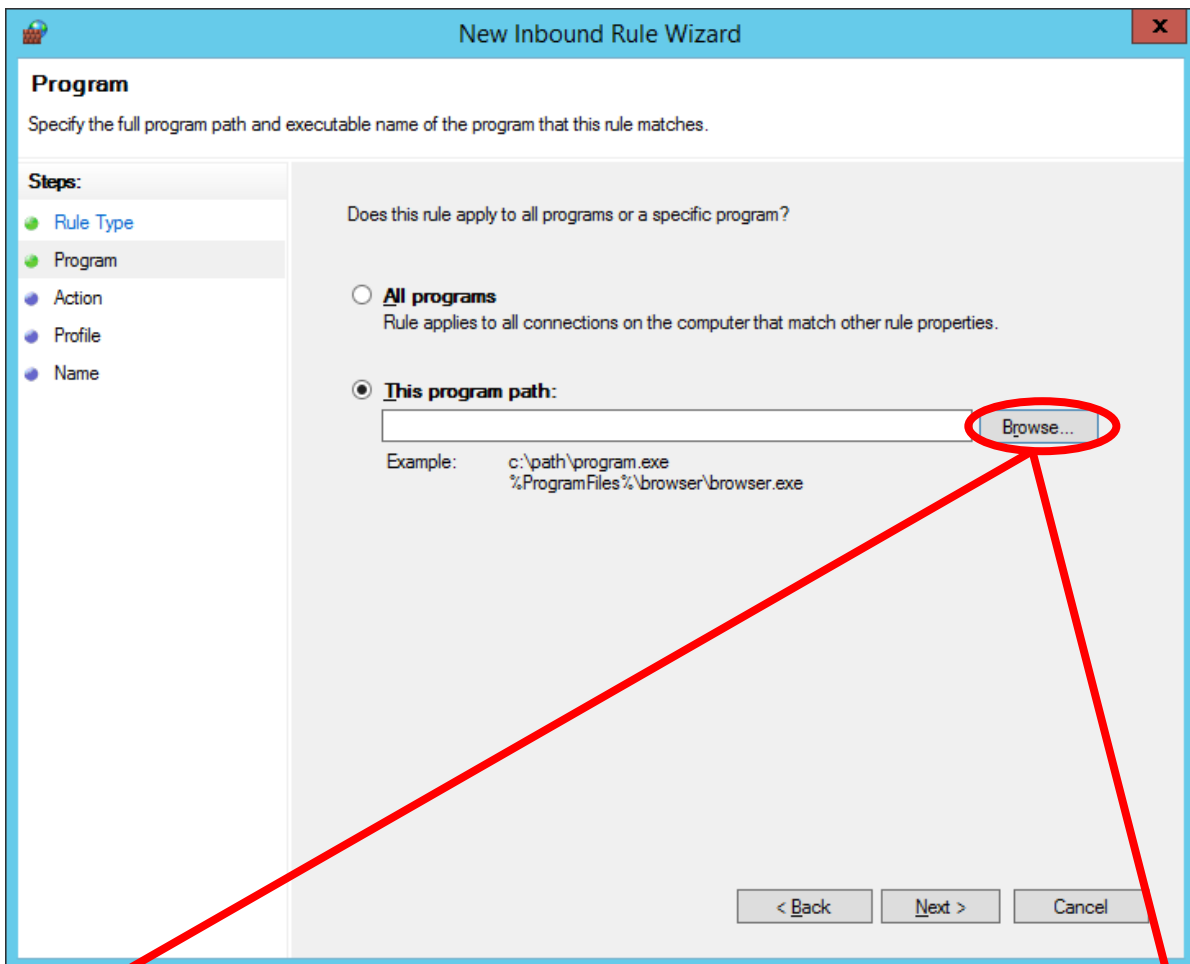
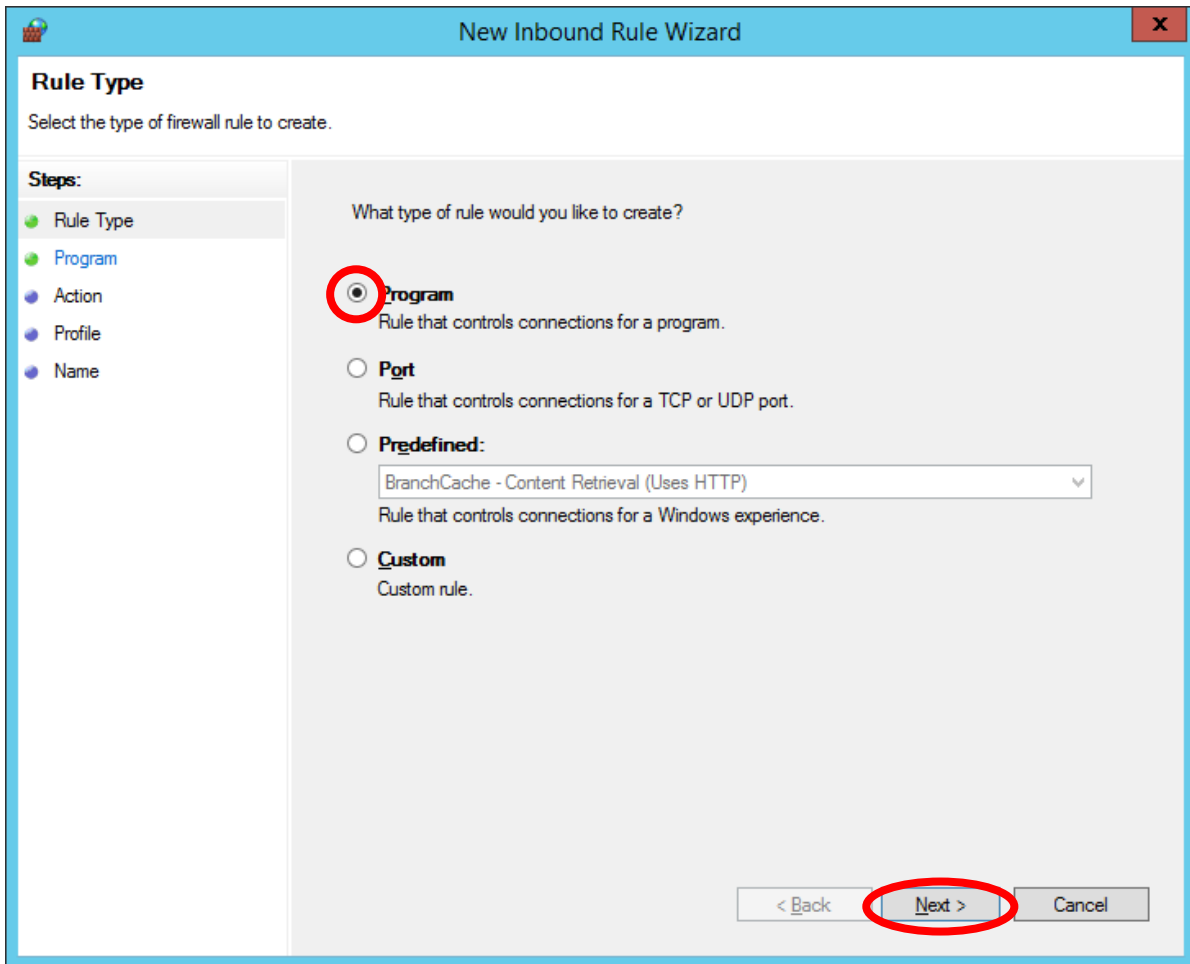


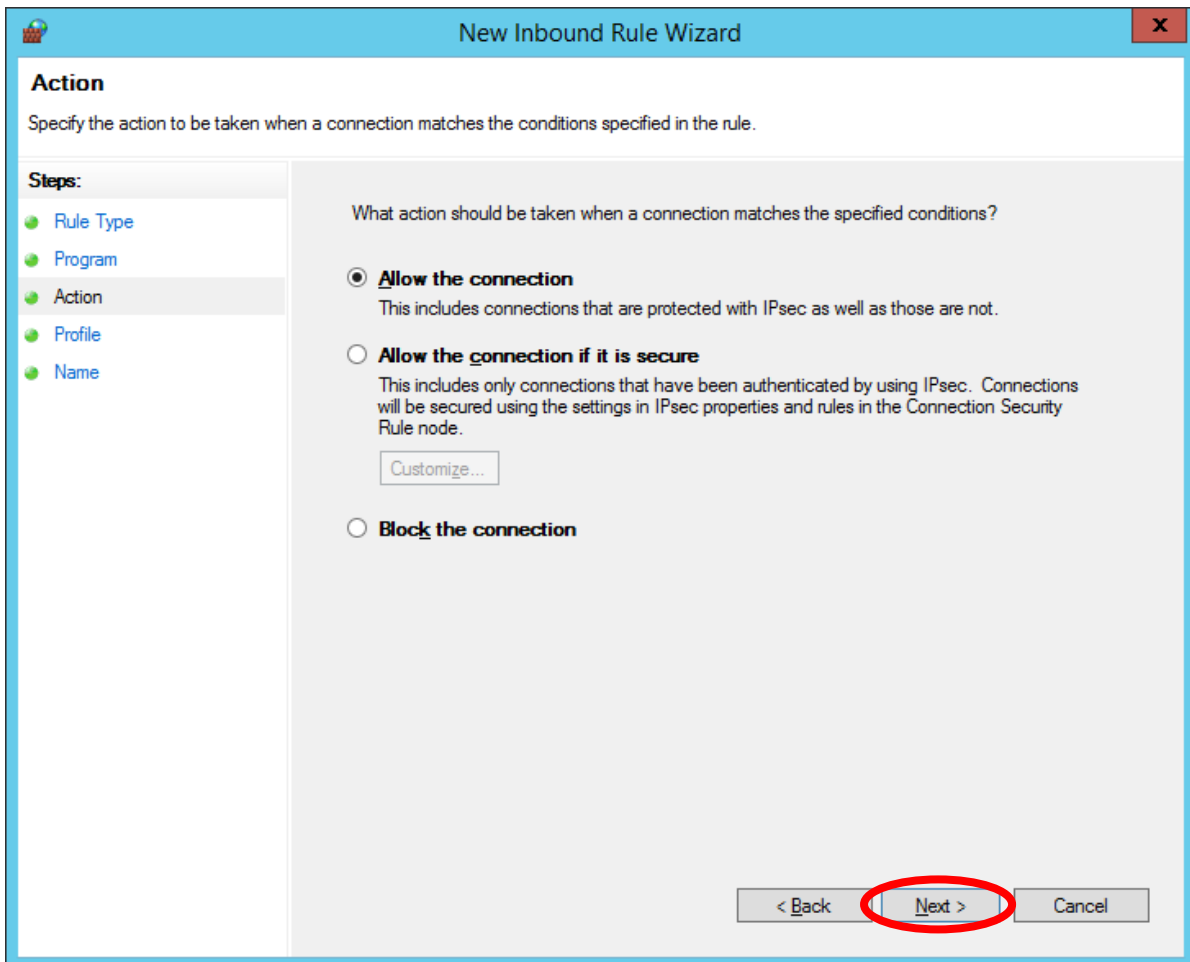
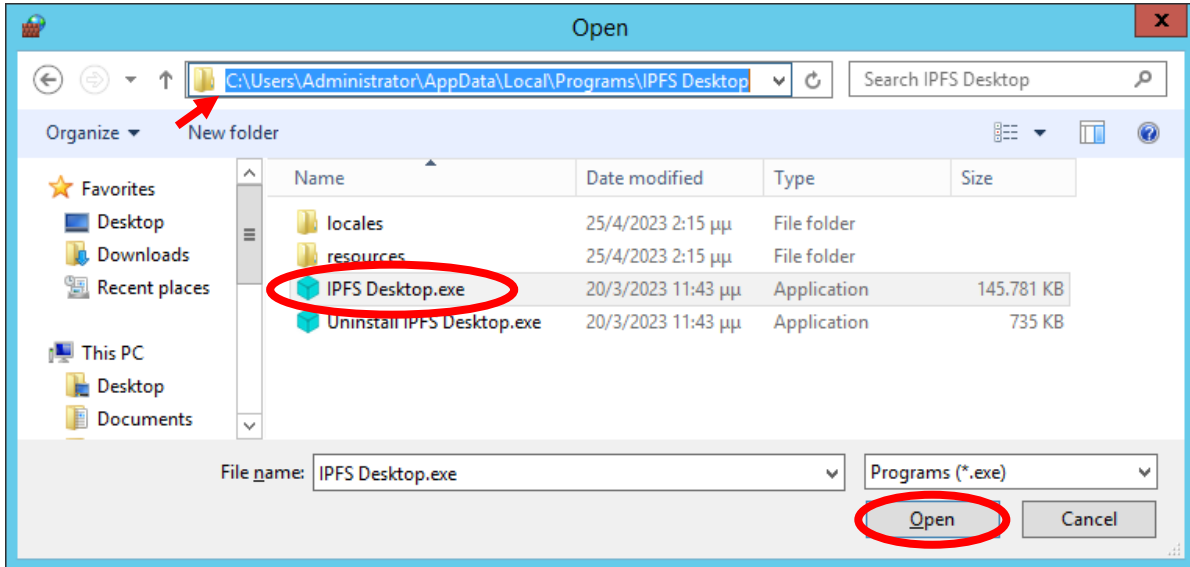


Στον firewall δώστε δικαιώματα εισερχομένων κλήσεων στην εφαρμογή IPFS desktop.

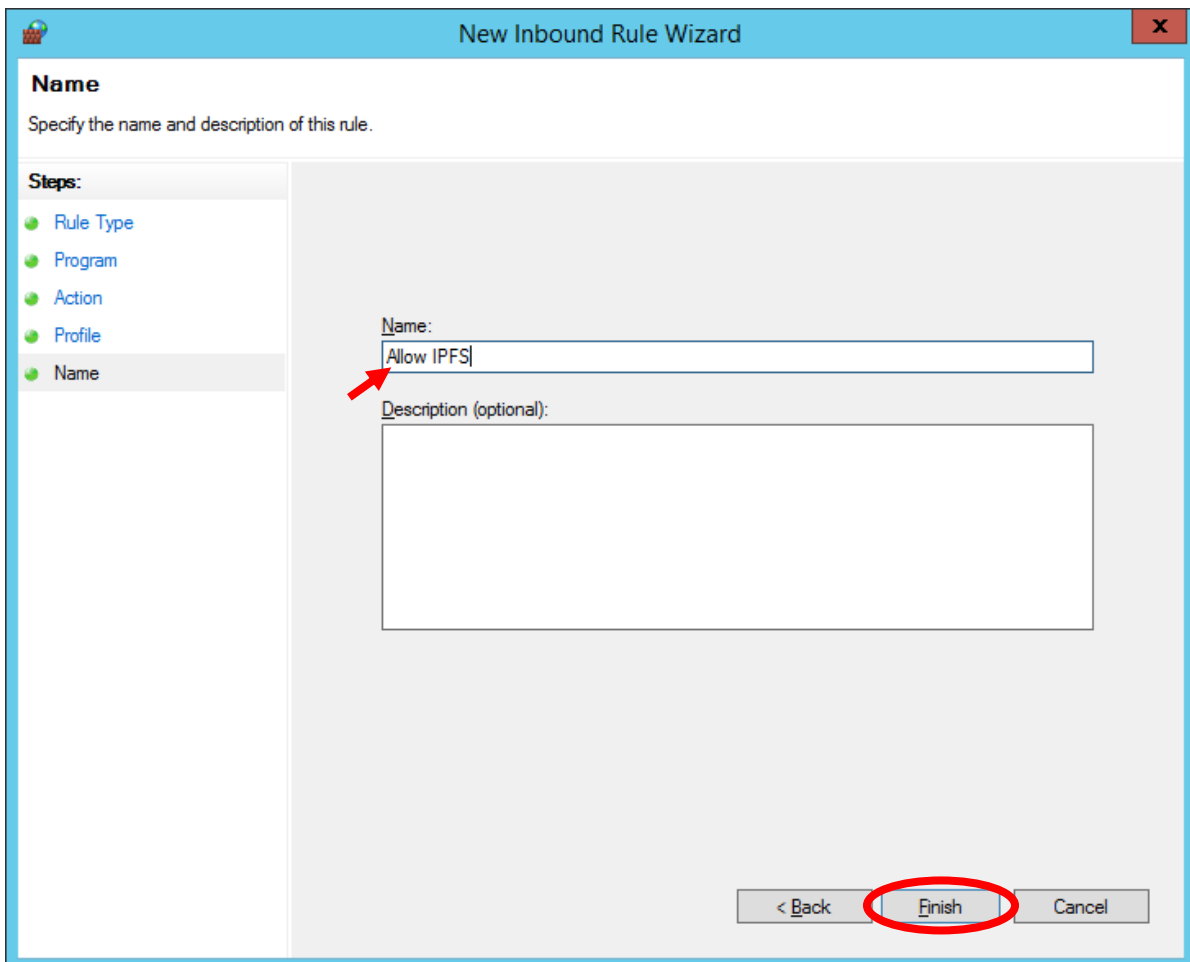
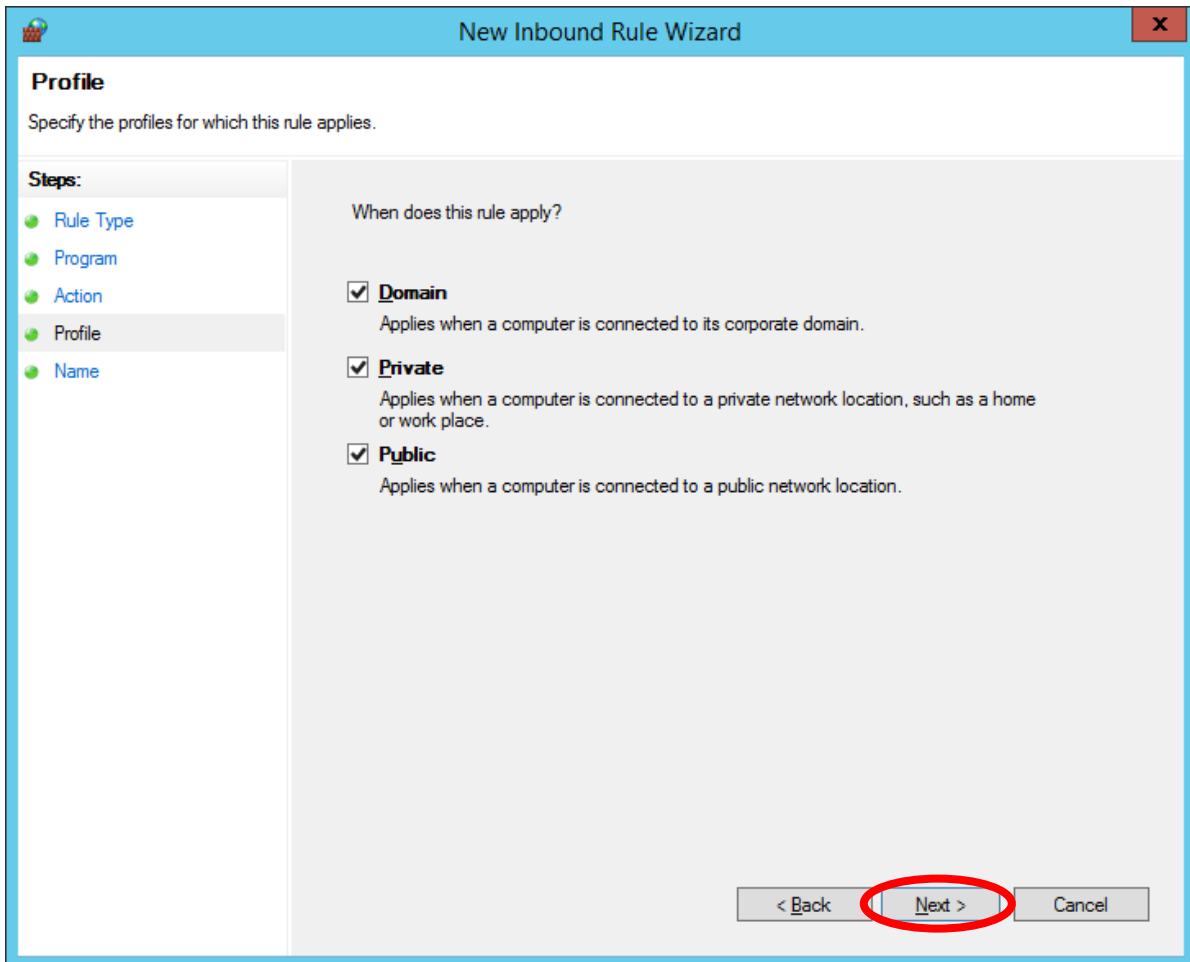












Η σελίδα σας θα βρίσκεται στη διεύθυνση:

<https://ipfs.io/ipns/k51qzi5uqu5djb14ho94ec8hnhzhoiz75rmfc0s234djt5c1325sz1o4ltrfer/>

Προσοχή! 1) Συχνά υπάρχει μια μικρή καθυστέρηση μέχρι να είναι διαθέσιμα όλα τα αρχεία του φακέλου. 2) Αν ο υπολογιστής σας μείνει εκτός IPFS για μεγάλο χρονικό διάστημα, οι σελίδες σας θα διαγραφούν από τους peers.

