



*Full credit is given to the above companies including the OS that this PDF file was generated!*

### **Red Hat Enterprise Linux Release 9.2 Manual Pages on 'orgs.7' command**

**\$ man orgs.7**

ORGS(7) ORGS(7)

NAME

orgs - Working with Teams & Orgs

Description

There are three levels of org users:

1. Super admin, controls billing & adding people to the org.
2. Team admin, manages team membership & package access.
3. Developer, works on packages they are given access to.

The super admin is the only person who can add users to the org because it impacts the monthly bill. The super admin will use the website to manage membership. Every org has a developers team that all users are automatically added to.

The team admin is the person who manages team creation, team membership, and package access for teams. The team admin grants package access to teams, not individuals.

The developer will be able to access packages based on the teams they are on. Access is either read-write or read-only.

There are two main commands:

1. npm team see npm help team for more details
2. npm access see npm help access for more details

Team Admins create teams

? Check who you've added to your org:

npm team ls <org>:developers

? Each org is automatically given a developers team, so you can see the whole list of team members in your org. This team automatically gets read-write access to all packages, but you can change that with the access command.

? Create a new team:

```
npm team create <org:team>
```

? Add members to that team:

```
npm team add <org:team> <user>
```

Publish a package and adjust package access

? In package directory, run

```
npm init --scope=<org>
```

to scope it for your org & publish as usual

? Grant access:

```
npm access grant <read-only|read-write> <org:team> [<package>]
```

? Revoke access:

```
npm access revoke <org:team> [<package>]
```

Monitor your package access

? See what org packages a team member can access:

```
npm access ls-packages <org> <user>
```

? See packages available to a specific team:

```
npm access ls-packages <org:team>
```

? Check which teams are collaborating on a package:

```
npm access ls-collaborators <pkg>
```

See also

? npm help team

? npm help access

? npm help scope