

# tildeverse zine dist/issue-1.pdf

## Contents

<b>Diving Into the Depths Of the Tildeverse</b>	<b>1</b>
Intro . . . . .	2
~tel . . . . .	2
TriviaBot, idlerpg, And Other IRC Games . . . . .	2
Tilderadio . . . . .	2
Gopher . . . . .	2
Tildeverse Search Engine . . . . .	3
Tildeverse UUCP Network . . . . .	3
Cosmic Storytelling . . . . .	3
.tilde Domain Names! . . . . .	3
Tildenet . . . . .	4
Dungeon Crawl Stone Soup . . . . .	4
Minecraft and Minetest . . . . .	4
Various other games . . . . .	4
And more! . . . . .	4
<b>Being Social With Git</b>	<b>5</b>
Intro . . . . .	5
Intro to Git . . . . .	5
Git Collaboration . . . . .	5
<b>So, You Wanna Make A Tilde?</b>	<b>7</b>
Why? . . . . .	7
How? . . . . .	7
What? . . . . .	8
So, who actually did this? . . . . .	8
What Not To Do . . . . .	9
Conclusion . . . . .	9

## Diving Into the Depths Of the Tildeverse

author: ubergeek

## Intro

There's a whole lot going on with the tildeverse, and lots of little projects, many of which take some discovering. Here's I'll dive into some of them, which I've since discovered.

## ~tel

~tel isn't quite a tilde, but is still considered a member of the tildeverse. There's no shell accounts to log into, *but...* There are SIP extensions. These are basically phone extensions, in a small phone network (Yes, it's a network, as there are interlinked phone systems now on there). You can see what is going on by getting into #tildetel on IRC, and by checking our tilde.tel

## TriviaBot, idlerpg, And Other IRC Games

There's a few IRC games running. TriviaBot runs in #trivia. This bot asks simple questions, and it keeps a daily, weekly, and monthly score (As well as all-time scores). No need to register.

idlerpg is a role playing game, probably unlike any other you've played. You advance by doing *nothing!* That's right, the longer you idle in the channel, the better your progress. You get penalized for chatng, nick changes, etc. It's available in #idlerpg

BitBot runs in many channels, but in #meta, you can choose to be a ducc fren (aka, A Duck Friend, it's not a typo, you'll see that a lot in #meta) or a duck foe (Just what it says), but "bef"riending, "boop"ing, and "trap"ping them when they appear. If you see this in #meta:

```
\_o< QUACK!
```

You can type ",bef", ",boop", or ",trap". bef and boop are synonyms, and trap traps the duck. BitBot keeps track of frens and foes.

## Tilderadio

The tildeverse has its own radio station! You can join in with the fanbase in IRC in #tilderadio, and listen in at <https://tilderadio.org>. People run shows, and often interact with the audience in IRC.

## Gopher

Most tildeverse servers also host content via the gopher protocol. You can access this using `lynx` on the shell of most tildeverse servers, or by using the tildeverse

gopher proxy. All of them are also linked on the tildeverse gopherhole (Yep, that's what gopher pages are called) at [gopher.tildeverse.org](http://gopher.tildeverse.org). Check it out, there's dozens of us! Dozens!

### **Tildeverse Search Engine**

The tildeverse also has our own search engine, available at [tildeverse search home page](#). There is a Firefox plugin available in the Firefox app store that will set your browser to use it as the default search engine as well, written by Yours Truly (ubergeek).

### **Tildeverse UUCP Network**

Way back when, if you wanted to transfer files, or send email, you used the UUCP protocol (Unix-Unix Copy Protocol). Some tildeverse servers have this set up today. You can join in [#uucp](#) for more details. Praetor is wrangling that project, the site is at [dataforge](#).

### **Cosmic Storytelling**

[Cosmic.voyage](#) is a tilde server, mostly... But with a very central focus: Writing stories. Here is an excerpt from their site:

```
Here are the annals of Relay Station 001, housed at Lagrange point
L4 in the SOL system. Found within are the records of
transmissions from the oldest and farthest flung ships to have
left Earth of old. These are pen-pal letters, reports, updates,
cries for help, and calls into the darkness. What is delivered to
the system is unfiltered. Some of it is intelligible. For some,
common language has drifted too far for clear translation.
```

So, that's the gist of it: extra-solar travel email :) It has a very long, and twisting story so far! They are in IRC as well, in [#cosmic](#).

### **.tilde Domain Names!**

The tildeverse also has their own top-level domain name system: [.tilde](#). In order to request a domain name there, you can go to [#.tilde](#) in chat, or do a pull request to the repo at [tildegit](#).

Now, these aren't domain names you can hit everywhere, sadly. More an experimental feature. You can however use one of their resolvers for your DNS server, and then be able to access [.tilde](#) domains. Their resolvers also enable you to access regular domain names too. Many tildeverse members use these

resolvers, too, so you hit those .tilde domains from most tildeverse servers. How to do that, and more can be found at the tildenic page.

### **Tildenet**

A number of tildeverse members are also linking their machines together via VPN connections, with their own address space, which allows the servers to offer far more services openly on that network, that wouldn't be safe to run to the wider, open internet. Things like remote file systems, telnet games, etc.

### **Dungeon Crawl Stone Soup**

Many of the tildeverse members have a command "dcss", which is a wrapper to just ssh to dcss@crawl.tildeverse.org, so you don't really need to be on a tildeverse to use it, but it's a lot more fun!

### **Minecraft and Minetest**

There are a number of minecraft and minetest servers out there, like theasylum, thunix, and team are some of the minetest servers. town and team host a minecraft server as well. The "Unofficial Tildeverse Minecraft Server" has a page here which has a map, list of players, and link to the mumble chat (A voice chat system).

### **Various other games**

Other games that aren't as popular (Yet), but still are out there. Thunix hosts a bzflag game server. Most servers have the bsdgames package installed (Or equivalent), which has atc (An air traffic controller sim), tron bikes (From the movie), adventure (Text-based rpg), monopoly, and gofish. The tilde's MOTD should show you those on login.

### **And more!**

There is way too much to cover in a single article, but some servers host tor proxies, forums (like iris and bbj), and other services. Hopefully those are documented on the servers themselves. If there is a service that I didn't cover here, and you think it should be, hit me up at uber geek at thunix dot net, and I'll do a follow up :)

# Being Social With Git

author: ubergeek

## Intro

One underrated social aspect in the tildeverse is collaborating via code or document creation. Many people consider that to be “loner work”, when in reality, it can be much, much more!

A great way to socialize while creating content such as source code, or documents, is via git. Git is generally used for source code control, but it can be used for any arbitrary non-binary formatted file: To do lists, calendar events, or just plain text files, such as this one.

## Intro to Git

Git is a command-line tool, available for most operating systems, and installed on every tilde server (Every one that I know of at least), and there’s a reason for that: It is a very powerful tool, used for managing configuration files, source code for projects from the web (Github is one of the most visited websites on the internet, as an example), and many other reasons. ~team uses git to edit their website, even!

Git works by creating a database of sorts that tracks changes made to all files inside of a “repo”, or “repository”. A repository is just a directory of files, with the git database files inside of it. One of the best features of this system is that anyone who “clones” a repo (More later) has a complete copy of the entire history of the project! Very resilient to things likes server outages :)

The command “git”, by itself, doesn’t do much, aside from spitting out some help info (Which is useful information). It works in phases: Edit file, commit change locally, then push change to another place (Usually a shared location for many people, like TildeGit).

There are many front ends to git. TildeGit uses a package called “Gitea”. GitHub uses their own. GitLab is another one. There’s actually quite a few.

## Git Collaboration

There’s a couple of ways you can collaborate with git. One way, is that each user has a copy of the git repo locally that they work on. When one person is done working, they “commit” their changes (This means they write all their changes to be tracked in their local copy). Before starting work, they “pull” the other person’s changes into their local copy, and then work.

This works well if there is only one other person working on the project with you. But, what if there are 3? Or 10? Or 50? It easily gets hard to keep track of whose copy you've pulled in. In this case, most teams have a central place they store their repo, everyone pulls from there, and when done with their local changes, pushes their changes.

Another neat feature are "branches", where you can stage changes to be merged in later, and not have to worry about stepping on other people's changes until you're done.

If you want everyone to be able to propose changes, but with you (Or a team) controlling what gets put in as "the one truth", others can propose a change in a local copy of their own, which you can review before merging in their change.

Gitea makes most of this process easier by calling the process a "Pull request". By performing a pull request (A PR), Gitea forks the repo to a local copy owned by the user doing the PR (You, presumably), where you can make you change. This also opens a special ticket called a pull request ticket.

From the shell, the process looks like this:

```
git clone https://tildegit.org/ProjectX/neatX.git
git checkout -b FeatureY
vi ./module/Zimm.sh (Editing the file ./module/Zimm.sh)
touch ./subModule/NewFile
```

This has your changes made, but if you do this:

```
git status
```

It will say there's nothing to do, but you have untracked changes. `git status` is very helpful to show you what you need to do, in order to get your changes done. Let's commit these locally:

```
git add ./subModule/NewFile
git commit -am "FeatureY changes, ready!"
```

`./module/Zimm.sh` already existed, so you didn't have to do anything, but commit. `./subModule/NewFile` was new, so we had to add it to the repo, to be tracked.

So, now these changes are in a branch in your local copy. If you want someone to review them, you need to give them a specific location, and way to pull your "remote" in (A remote is a copy of the git repo, that you don't own). Let's say I'm on `~yourtilde`, and your copy is on `~team`(Your `~team` login name is `FeatureXUser`), and I (ubergeek) want to pull it in to merge (I own the repo):

```
git remote add -f FeatureXUser ssh://ubergeek@tilde.team:~FeatureXUser/repos/ProjectX
git fetch FeatureXUser
git merge ..FeatureXUser/FeatureY
```

The above adds your copy as a “remote” for me, fetches all of your changes and branches, and then merges in your FeatureY branch into my “master” branch. This means all the changes you made are now “live”.

Gitea does a lot of this work for you, in a web-based UI, and does some more legwork (Creates the ticket, notifies the repo owner or owners of an proposed change, etc etc)

## So, You Wanna Make A Tilde?

author: ubergeek

So, you want to make a Tilde server (aka a pubnix server)? You’ve got time on your hands. You want to make a new and cool thing, right? Awesome! This article is going to step you through some of the pitfalls, so you can try to ensure it’s going to be fun.

### Why?

Before you start down the road, ask yourself “Why?” Why do you want to make a tilde server? To be ultimate lord over a domain? To have a cool art project to play with? Any reason is a valid reason, but the “Why” answer will help you build your tilde, and cultivate it’s community.

### How?

Yep. “How?” Running a tilde server is not free, money-wise, or time-wise. It’s work, and money.

Are you cash-strapped, but have an old machine and somewhat stable internet connection? Maybe home-hosted is the way to go. But, where you save on money, you generally spend more time on it. Self-hosted can be fun, but you’re going to spend more time on administering it, since you’ll be responsible for all of the hardware and software up to your internet connection. If the hard drive dies, or a RAM module gives up the ghost, it’s on you to replace it, and your tilde will be down until you do so.

Are you time-strapped, but extra cash? A VPS is a fine choice for a small tilde, and can be relativeley cheap. You can get a small VPS for \$2.50/month US, and it’d be suitable for at least a dozen users, who aren’t hammering on the machine. There’s also dedicated servers (You manage the whole machine, hardware and software, your provider manages power and network, but you don’t own the hardware, and they replace bad hardware), or you could co-locate them (You are responsible for everything but power, and a internet drop). Each option

has its own pitfalls and benefits. It boils down to the more you pay more for resources, the less you are responsible for.

What OS? Lots of tildes run Ubuntu, so help is plentiful there. There's some BSDs, as well, and at least one Redhat-based tilde. The more unique your distro/OS, the more you'll be on your own. Something to keep in mind there as well.

One thing to keep in mind, though: If you're going to do the same thing every other pubnix operator does (Most commonly used distro, most common web server, most common forum software, most common set of service, etc; then what will make your pubnix different? The "Why?" answer will fill this part in, which is the reason why it's the most important question, and comes first.

### **What?**

I know. You're saying right now, "What does he mean 'what?' A tilde you daft bastid!"

The "what" defines what kind of community you want to build. Are you just building a clubhouse for close friends? A living digital art community? A community of arduino hobbyists? This is a key thing to figure out BEFORE you decide to launch your tilde. This also guides your technical decisions, and things to install, services to provide, etc etc

### **So, who actually did this?**

I can tell you from personal experience with thunix, that I went through these same steps. Some questions were pre-answered since I essentially took the reins from an already existing project.

Why? I felt I had a lot of experience to lend to the community. My professional job is running, managing, deploying, and maintaining linux servers. Some of them are internally pubnix boxes. Users use them for building code, doing git work, etc etc. Some things are missing, like chat, but there are other tools for that sort of thing. But, nonetheless, experience I have, and decentralization is what I wanted to bring. I also wanted to give users a place to play with what a CI/CD environment is like, and as such, we designed thunix to be built and maintained in a devops manner: Users and administrators work hand in hand to build the system.

How? My job pays well enough that I have some disposable income, on a regular basis. Not a starving artist, or a contractor wandering from job to job. Also, I had some cloud stuff already, and this seemed like a good enough reason to get some dedicated hardware, and consolidate, and save some money. Also, the project I took over for suffered from Bus Factor of 1. The users already lost everything once, and I endeavored to ensure that would not happen again. I



settled on git-controlled system configs, and distributed backups (Machine backs home up, and admins are responsible for copying that backup off regularly). Anyone from the admin team can get everything running in a new place, in short order. Any one user can replicate what we've done, shy of the backups restoration (Access to the backups is a root-only job).

What? While old-thunix was big, and offered lots of services, to me, it seemed like it lacked a community. So, I decided to turn it into not just a service provider, but a provider for those who are technically inclined, or willing to at least put in sweat equity. ie, I'll never reject a pull request unless it endangers stability, or security. I'll also step people through contributing, if they need.

### **What Not To Do**

Mostly, the one thing you want to avoid is overextending yourself. Taking on a community is a big deal, and not to be taken lightly. Your community members put their trust in you, and if you cannot sustain the community from your own resources for a period of time, you need to be upfront, and scale down your expectations appropriately.

There is nothing wrong with NOT making your own pubnix, and just using the resources freely offered by other folks.

Don't jump foot first into it, without planning a little. Many people have done this before you (They used to be called "free shell providers"), and many will happily lend a hand, or let you bounce ideas off of them. And, popping up a monstrous server, expecting thousand of people to swarm into your clubhouse right away is probably not so great an idea, either.

Don't be afraid to start small. Most pubnixes offer a shell + some public webspace. Don't feel like you need to start off right away with more than that. You can incrementally add things. For most things? 2 cores, and 4GB of RAM is plenty for a user or two, offering only that. With cloud providers these days, it's usually pretty easy to scale up. Not so much to scale back down.

Don't ask people how you should implement something. I'm not talking about troubleshooting tips, we all need that from time to time. But asking, "What should the default shell be?", "Which web server should I use?", "Which Minecraft server should I deploy for users?" are bad questions, in some contexts. You're in charge of the community's resources. You should be deciding how to offer what. A big part of starting a pubnix is architecting it.

### **Conclusion**

I guess to conclude, you may not agree with these steps. That's fine. The great thing about these sorts of projects is that you can do you. It's a big job, or as big as you want to make it, but very fulfilling. Don't bite off more than you can

chew, and don't worry about making mistakes (Even ignoring this doc), and have fun doing it!