Sysadmins Information

Do you plan to install a Tiki or you already have an instance? In both cases you will need some basic context regarding your requirements, environment and what it means to admin a Tiki.

Don't have a Tiki website already?

- For information on trying a Tiki demo, go to Try Tiki.
- To get the Tiki package, see the information at Get Tiki.
- If you installed Tiki but you have problems, please see the sources of help at Get Help.

What is required to run a Tiki? First of all, a person.

Congratulations and thanks for your interest in Tiki Wiki CMS Groupware. We hope Tiki will meet your needs for a highly configurable web platform. If there is anything we can do to help, be sure to let us know.

The requirements page is quite explicit about what you need in order to run a Tiki instance https://doc.tiki.org/Requirements . So we hope that any system administrator will be able to find a proper environment to run Tiki. It really isn't much to it, at its core Tiki is (very) simply put a PHP website, so it will happily run where your other web applications run. This being said, administrating a server for Tiki or other web applications is a job in itself. We will try to provide here a birds eye view of what is expected of you. But if you are a total beginner in this field of system administration, you should first:

- take the time to learn a bit about the different web hosting technologies;
- try to mimic certain recipes available online, though you should in time always make your own decisions and come up with your personal way of providing a Tiki to users;
- have patience but also have a bit of courage, this means having the inner desire to perfect your skills is an advantage, but always with an eye on the future, always eager to try new things. Passion helps here, a lot;
- be ready to become a tougher individual, ready to deal with the 24/7 stress of running any service others depend upon, more so with all the bad stuff happening online. This is no joke, you should expect problems, in fact problems will become your very job: either from your software/hardware stack, by your own hand or your users, and also via the armies of crackers (not *hackers*, please learn the difference) just waiting for you to have something, anything, online. Down to your electricity and/or internet providers, whom will become your archenemies if building your own infrastructure. Heat dissipation too;
- get in shape to build trust around you. A sysadmin that doesn't have a good moral compass is a danger to himself and others. Knowledgeable admins can be easily found. Trustworthy ones, at a level that people will hand over to you their personal info, sensible files, even their family's well being and the future of their company - that kind of person is more rare. Some sysadmins might even be under the impression that they are wielding some sort of power; this will trigger all the red flags one can imagine.

Let's go through your technical options now and also a few other points you might want to consider.

Shared hosting and cloud VPS

So where to host? The popular option is to either use a shared environment as the go to "pay and play" or rent a VPS, from a provider.

- In the first case you will not have to deal with the intricacies of having to set up a whole server, and everything is ready for you at the smallest price compared to the other options. Usually, just don't pick the cheapest offer out there, because it will have hidden costs in the terms of functionality, resources and availability. Simply research your provider, get the most for your money, and just try it out as there is not much administration involved. And it is standard to be able to scale your resources, so you can easily increase them if needed.
- Only in the second case, you will "elevate" yourself to a true sysadmin as it will be required of you to set up a server for your services in a virtual machine provided by a third party. This is a good option if you or your organization, don't have the resources, you aren't interested in running you own servers, or you just need to test your hand. Keep in mind this option is feasible only in the basic to mid tiers - prices will climb up quickly if you need more resources. This is the point where you should calculate how much will it cost you in the long run.

Roll your own. Containerized, virtualized or bare metal?

If your organization already owns the infrastructure or they intend to build one, you are not alone. In the cloud era, there is also a resurgence of the need to own your data, for privacy reasons mainly but also costs, if you have needs exceeding what is offered by subscriptions for some cloud VPS instances or even renting physical servers. There is also the case when you need a Tiki instance strictly in your LAN, for example when building an intranet portal; the choice to self host is imperative in this case. A few points here.

- Members of our team are making good use of container technologies like docker and podman, but mainly for developing and testing; it is not something we see often in production. It can be used just fine but in our experience the organizations are using containers for Tikis less often than the next two scenarios.
- Virtualization is the the preferred option for setting an internal but also a public facing Tiki. Popular virtualization technologies like KVM and Xen, or proprietary solutions like VMware have no problem with a common web server + php and some SQL databases in a VM. And of course they are ubiquitous in modern corporate IT landscapes, so it makes sense to use your existing infrastructure. We urge admins to build their solutions using virtualization if they didn't already, for security via segregation, ease of restoration and consolidation purposes to say the least.
- Running on physical hardware is required when no virtualization is in place because you might have older servers that do not worth the effort or they do not have the proper CPU instructions sets. Or when you plan for a monster of a Tiki that really needs all the resources and the lowest latencies possible. This is a return to the old mantra, and you know what to do if you aim that big.

GNU/Linux is the best OS for Tiki

It doesn't matter the distribution as long as you have a Tux under the hood, and there is no point in even mentioning the other two major proprietary operating systems that have a bigger (desktop only) market share. Tiki builds on the same freedoms and philosophy with this family of operating systems, being in fact one of the many technologies they spawned. We don't see any problems if you want to host your services on a rolling distro like Arch or Tumbleweed, bleeding edge ones like Fedora or the distros for purists, like Trisquel.

Though in production the general consensus is that you should opt for "stability" as in a slower paced, maybe sporting LTS versions, with mature packages like Debian, derivatives like Ubuntu, RHEL or rebuilds like Rocky Linux or AlmaLinux. You can't go wrong with any distribution when running Tiki, so it is up to you which one you choose, depending on other considerations and even personal views.

Here you can find comprehensive information about installing a Tiki instance in different environments https://doc.tiki.org/Tiki-Installation-Guide

First things first

As soon as you get your instance installed and running, you should check the following:

Backups

Yes, please start with backups. Weird thing to say to a system administrator, but everyone needs a reminder that losing data is not acceptable, at least not all the data. As soon as you manage to have a Tiki running or anything for that matter on a new server, you should start with making some backup decisions and put some automation in place from day one. Sooner or later this will spare you a lot of grief and it is only a matter of time before something happens, so it is not a question of "if" but a question of "when". Here is some info about what you need to do at a minimum https://doc.tiki.org/Backup which means having a database dump and your files copied somewhere else.

So think about your backup philosophy (if you don't have one, make it up right now):

- what are your Recovery Point Objective (RPO) and Recovery Time Objective (RTO)?
- respect the golden 3-2-1 rule: have at least 3 copies of data, on 2 different media, 1 copy being off site; it is acceptable that one of those might be the production data itself;
- more planning: how much space do you need for backups, where to host them, how long it will take to create a backup but also transfer it via the network or the internet, and if the backup does not succeed what exactly happens?
- full vs. incremental backups; as a rule try to have as many full backups as you can; and please do not forget about encryption if you don't trust the destination environment;

- review your settings and test the backups later (by restoring them), if possible regularly, despite being regarded as a tedious work.

Check the health of your Tiki

You should check the health of your Tiki right after installing, but also once in a while and even more so in case of problems, via the **Tools > Server Check** option in your Tiki's control panel. The usual culprits will be clearly presented and explained, with meaningful hints. Great tool, you should use it.

Permissions

Permissions issues are not common encounters in shared hosting as you don't manage much, your host does, and so it is their job to keep them in good order. But they do happen when you are in charge. Their effects can be quite noticeable in any deployment starting from weird Tiki behavior, explicit errors in control panels or pages and ending with a complete website crash. Though when you get them right, there is nothing more to it.

So problems rise due to user errors, a misbehaving script or inappropriate defaults in the OS/specific software servers. Never use full 777 permissions for anything and never run anything web as root, not even when testing as you will learn the very wrong lesson that they are a quick fix. Use at least 755 for folders, 644 for files, owned by the proper web user.

By the way, why do you think you are seeing in logs scans for "/test", "/dev" or brute forces for "admin" with password "123456"? Because people use them in testing, not production. The same applies to your permissions, even when testing.

Resources

For sysadmins it is a good idea to first over provision and second to have an alternative, a different environment. You should never get 2 cores if that is the bare minimum for a web server with a few other services, as you should never try to get away with 2GB of RAM or 50% more of your current storage needs. Remember, in the long run requirements will always go up, and the moment resources will count the most is when... you need them or you have problems and you don't have them; more so, you may invite downtime and other sorts of issues when trying to resize or add disks/partitions, RAM or cores.

We are not recommending wasting money here, but remember, if you are cheap that is exactly what you will get and in an organization it is not your job to complain about the IT budget being big, but rather to get what you need for your job. Probably there is someone else in charge with getting the money you need for the setup, and that would be an inappropriate thing to do, transferring this burden to a sysadmin. That is saying: you should find proper resources for a job well done not try to squeeze yourself into "some" resources. And if something is critical it will cost more to keep it in good order, so

everyone should treat the corresponding expenses as such. It is not uncommon nowadays to think about loosing your data in terms of: how much does our data/solution worth? Well... as much as the company itself, even more if you affect other parties.

We were mentioning you should also have an escape route in case of problems, being a second server, a second VM, a second shared hosting account (ideal, triple that and combine). This means identifying and getting resources proactively for when the main environment will go down. Because it will just go down at some point. Hybridization is good here: for example if you have a local physical server, you should also buy a VM in the cloud (or at least become very fast to get it and set up one, ie all decisions taken, scalability in mind, have some reserve money, accessible DNS services with low TTL etc).

For the self hosted hardware tinkerers out there: you already know that you should have some spare parts for pretty much everything, so we thank you for being an inspiration for us all.

Monitor as much as you can

Everything is good and running, but we keep telling you that something bad is going to happen. How will you know?

- first, you should set-up some external monitoring, because you will need to know if something goes down and local monitoring does not help: for example if your server will crash or doesn't have a connection anymore, it will take down the email service and it will not be able to alert you. This can be a simple script that just pings or curls machines and services from another server, a free or paid external monitoring services that checks your uptime, or even starting a dedicated machine that will check for problems, with tools like Zabbix, Nagios, etc. This is mainly for the complete loss of connectivity to the server;
- second, some local monitoring is a must and the most info can provide at a glance, the better. Because maybe you will have only a couple of services go down but not all, some performance issues, software bugs, you want to keep an eye on the load etc. Local tools like Netdata will be invaluable (Zabbix too, via its agents), and if those can also send out an alert, that would be great. The web hosting control panels, like Virtualmin, do have mechanisms to get you notified in case of problems;
- third, the users are a sensor net by themselves. Obviously people will try to contact you if the services are down. So keep paying attention to what they are saying, learn to communicate with them in layman's terms, and also identify some quick checks that might help you understand things, from their side, from their perspective. Most of the times these alerts are local issues, misunderstandings and such. But you should check each and every one of their claims, and encourage them to keep reporting, but properly, tuned, maybe come up with a procedure. You will be amazed how much value an educated user can bring you when they already provide you their external IP address, give you the proper link, and they already checked themselves against the blacklists.

So drop that angry sysadmin routine and remember - you are doing this for them and they can help. And if they are disgruntled, you will be too, so make them part of the team.

Caches and indexes

Tiki already has a caching system, so if some page does not display the most recent content be sure to empty the caches https://doc.tiki.org/Clearing-Cache and as stated there, remember to empty your browser's cache too.

There are also other systems that can help to speed up the delivery of your Tiki, like PHP's opcache as a minimum, or even the memcached and APC systems. Mod_pagespeed can also be an option but use with care and test properly.

One of the most important features in Tiki, and wikis in general is the fast or even instant filtering and search. For this to work well on a Tiki instance we use a unified index. OK, it is a simplification, but for you as a sysadmin it is important to rebuild this index regularly, at least one per day via a cron job. Know that there are multiple options for the engine, as explained here https://doc.tiki.org/Unified-Index-Comparison

Availability and uptime

A complicated subject that doesn't get the justice it deserves, as the availability of your services will define you and your work as a sysadmin. Being always on is hard and nothing can substitute experience here. Ignoring the previous advises will not do you any good, as the uptime of your services is the sum of everything you do, and the downtime is obviously the reverse. As in most cases there is no recipe for uptime; so in order to get there just make it your everyday objective.

Preventing IT hardware and workloads to go down is done via a simple rule: just do not go down so take all the precautions one can think of. But if you do, always have the best of reasons if you are the one directly responsible; or if it is an accident that will become a war story, be prepared with a save. Above anything else - investigate, have data, be honest and forth coming, so do not let the impression of a marketing department speaking.

Nothing more to say here, not without staring into the abyss, as every downtime is specific to the cause that created it.

Use a control panel if possible

This is as much about productivity and automation, as it is about usability for you and a few privileged users that for example need access to the root of their website. If you are a sysadmin, than probably you are not a very good designer, and to a lesser degree a developer. And in an organization you will surely have colleagues or third parties that will need access to your server. Point is - you will need to provide access and allocate resources, maybe even provide your users with a tool to manage their account.

Can you set everything manually? Yes. But can you do it with the highest consistency, best segregation and also the highest speed possible? Sometimes yes, usually no. And you will end up reinventing the wheel anyways.

This is where a control panel comes in handy as this specific type of software can assist both administrators and users to get the best out of their server. We don't intend to suggest here proprietary ones like Plesk or cPanel. There are a few very good and open source ones that will make your life so much easier, like Hestia, ISPConfig or Virtualmin (that we like the most). They can provide you the ability to create and delegate a domain on your server, complete with databases, DNS, email, SFTP/FTPS and others in a matter of seconds after you filled in some fields and clicked a few buttons. So give them the proper attention, we doubt even a terminal ninja is able to do that without a plethora of in-house scripts.

Tiki Manager and the WikiSuite project

The previous control panel section is also a way to introduce another two projects related to us that will make your Tiki sysadmin life much easier.

Tiki Manager is a CLI tool that can assist you with creating, importing, deleting, cloning and backing up Tiki instances. It can even check their health for example. It sports a soon to be deprecated separate web interface, that we are dropping because we managed to integrate Tiki Manager as a package in Tiki, such that an instance is now able to control other instances, even on remote servers, even by creating their domains if the Virtualmin web hosting panel is involved. This brings us to...

WikiSuite is an effort to build upon Virtualmin via a modified install script in order to provide you with the best setup for a Tiki server. This means that after running the script you will end up with a proper machine that is able to immediately host your Tiki instances in the best way possible. To this end we integrated Tiki Manager as a feature in Virtualmin and you can find it in your domain's left menu just like any other option. We plan to keep adding to this solution as to become a ready to go script that will a provide an organization with everything it needs.

Secure everything to the point of false positives

Running a web server facing the world it is fairly standard, though it has some hot spots, security being one of them. We see it more of a mix of separate skills, experience that will come with time and most important - best practices.

Your security posture can be helped by the following:

- read up on the different types of attacks, and the simple mitigations you can take; default configs are always just the starting point;
- realize that a successful attack on a service can have a broader impact, and can ramificate in others via elevations and correlations;
- obscurity never means security and even in an internal network you should always use authentication, firewalls and VLANS, proper policies for your use case and encryption;
- update your entire software stack when a security warning is triggered by upstream, in fact that is the only time you should update your (now) corporate level solution unless you need specific bug fixes. It will in fact also trigger feature additions and bug fix related updates. Chances are for example a nginx or ssh CVE could apply to your web server, but also for your router and even at home, for your reverse proxy used by Kodi or Home Assistant machines;
- backups are a must when dealing with destructive intrusions, snapshots are even better in case of various cracks and cryptolockers (they do not exclude backups, of course), so maybe try to use filesystems that support them, like BTRFS, ZFS or simply use those provided by LVM;
- never have a public facing machine without at least the default firewall always active and configured; leave only the necessary ports opened; we like nftables a lot, and Debian 11 is all about it;
- try to improve your firewall versatility by using tools like Fail2Ban or CrowdSec, that trigger rules when an IP is trying to brute force the different services, as those leave a trail in the logs;
- if you are behind a good router, always do the filtering at the edge, before it gets into the network; OPNSense and pfSense are good examples of just-enough-OS routers that even have packages like Surricata, Snort and CrowdSec lately;
- learn to use blacklists, those will clear your traffic and ease the pressure on your resources, there is no reason you should allow known, repeated offenders; use pfblockerNG on pfSense and the native aliases on OPNSense to grab them regularly;
- have a good password policy, in general, not only on your servers and please use a password manager, KeePass and BitWarden are gold; now... why not generate random passwords longer than

25 characters?
- don't let anonymous users do anything but read/access in any service you are running;
- learn to run tools like rkhunter, clamav, and even auditing ones; WikiSuite has these installed by default;
- restrict administrative access to certain IPs and/or VPNs; this will drop the possibility of external access even if they somehow get your root password (you are doing something very wrong if that happens, review everything and start from scratch);
- and of course it would be excellent if you could use 2FA for the very same reason. Tiki supports 2FA, Virtualmin too;
- SPAM is an issue, being email, in forums or comments; use SpamAssassin in combination with blacklists for email, CAPTCHA for your forms, and confirmations via email for new accounts - once again, do not permit anonymous users to post/fill anything, only registered ones;
- keep in mind that you will be dragged in security disputes even if everything was fine from your point of view; might be one of your users, almost always unknowingly providing access to a bad actor. In this case education and training becomes a must.

There is a lot more, but we hope this will get you started if you want to become your very own sysadmin.

Where to get help

Tiki is a bit complex, with many features and preference options, and some things may be done differently than in other similar software, so familiarization may take some time. Please visit the documentation site for details on feature use, etc. Also make use of the sources of help listed on Get Help, in particular:

- Ask questions in the Forums or in the Matrix chat (also available by clicking "Open Chat" button below).
- Get involved in the Tiki community.
- Please feel free to participate in our monthly roundtable meetings.