Tiki Manager within Tiki

Development has started and this will be part of Tiki25. Initial code: 
https://gitlab.com/tikiwiki/tiki/-/merge_requests/1374 Next up: Wiki Plugin (done) and Tracker field type

In the mean time, you can use Tiki Manager (CLI or web interface) or 
https://gitlab.com/wikisuite/virtualmin-tikimanager

**Executive summary:** Tiki Manager is very helpful but there are two families of limitations:
1. It is designed for a single admin (all or nothing) and thus can't be used for self-serve requests for demo instances.
2. It automates operations for Tiki management, but not server management (create web space, set domain name, configure SSL, etc)

With enhancements to Tiki Manager and Tiki, when used in conjunction with Virtualmin, this will permit in one Tiki instance to easily manage hundreds or thousands of other Tiki instances.

Why

To address these use cases (in order of priority)

1. Easy for Tiki consultants to set up a Tiki for a demo to a lead or a client
2. Provide end users an easy way to request a demo for a few weeks and won't get messed up by other users or reset daily, like http://demo.tiki.org/
   - Provide way more demos that are tailored for specific use cases (Easy demos of profiles, a clone of a nice template site, etc.)
3. Documentation: Easy access to Tiki instances of recent versions (including trunk) so as to write documentation, and collect nice screenshots and screencasts
4. The next generation of the show servers
   - Also a great use case for screenshots and screencasts
5. In a dev/staging/prod context
6. For Tiki hosting services, who can leverage Payment
7. A way to demo Tiki Manager (But first we need to add a way for Tiki Manager to install Tiki Manager recursively)
8. Continue on https://wikisuite.org/Orchestrator

How

Similar to what PluginCypht does for webmail, embed Tiki Manager's web interface in a wiki page with

- **WYSIWYCA** of instance list
- actions / fields hidden or shown via plugin parameters (to permit very simple interfaces)
So a GUI to configure a GUI with configuration saved in a wiki page 😊

Benefits

- Depending on the use case, we can create different wiki pages with different
  - permissions
  - settings
- PluginAlias can be used to make it easier to create many variants
- Converge energy/focus on Tiki Manager's web interface (which needs some TLC). Using the plugin without any filters will essentially show the full feature set like it does now to the admins.
- Can connect to n Tiki Managers
  - So you can make a unified dashboard of all your Tiki instances even if they are managed by different Tiki Managers.
- Users register to a Tiki and then can manage their Tiki instances
  - Can reset their password, etc.
- Low barrier to entry (Activate, fill in tiki-manager.php and use plugin) while offering lots of flexibility (undeeded features are hidden, support many Tiki Manager instances, additional field via trackers, etc.)
- Can be combined with Tiki Firewall or Two-factor authentication for higher security

Parameters

We are brainstorming options

Option 1

```{tikimanager(connections=abc, showactions="create|backup", hideactions="clone" hidden="tmp|path")}
```

Option 2

```{tikimanager(connections=abc, create=y,clone=n,backup=n, hidden="tmp|path")}
```

Option 3

How to segment sub-actions? Ex.: delete backups with backup section

```{tikimanager(connections=abc, create=y,clone=n,backup=n, hidden="tmp|path")}
```

Connections

- Pick from list of available connections to actual Tiki Manager instances
  - Could be managed in tiki-manager.php (more ideas below)
  - Tiki will one day install Tiki Manager via Packages, (PHAR) or similar
Actions available

Pick from any action available (create, clone, backup, etc.)

Instance fields

Each field (connection, name, URL, tmp, etc.) can be either normal or set to:

- Read-only: The user can see but not modify
- Hidden: Data needed to function but hidden to the end user to keep UI simple.

Default value overrides

The idea is to permit overrides from data from tiki-manager.php (see more info below)

Not sure yet about this one. Maybe it's better to just create an extra connection, like Tracker Tabular can have various import/export to a same tracker.

tiki-admin.php?page=tiki-manager

Activate the feature and related preferences

tiki-manager.php

Since we don't want some data stored in a wiki page, we can configure n connections

- Name
- Ip/domain/port
- Connection type (local, localhost SSH, external SSH, FTP)
- Default patterns
  - Ex.: If domain needs to be *.example.org or *.demo.wikisuite.org because DNS is set accordingly on the target server
    - Specifically, joe@example.org could request acmecorplegal.demo.wikisuite.org or acmecorplegal.demo.tikitrackers.org
- When applicable, offer Virtualmin options (Use/Create Virtual Server or Sub-Server, set domain name, etc.)
  - https://wikisuite.org/PHP-code-for-Virtualmin
  - Keep it open ended to support other APIs later
- Potentially create a new VM (ex.: Linode APIs)
- Perhaps one day, Tiki Manager will manage MultiTiki
- Pattern for database username and database name (end users should never have to enter these)

Tiki Manager data and tracker data interop

- All data in Tiki Manager should be available within Tiki
  - Reports, automation, etc.
- How do we permit extra tracker field per instance?
Perhaps we use similar to http://doc.tiki.org/User-Tracker ?
Do we map Tiki Manager fields to regular tracker fields? (A use case for system trackers?)
Pick tracker and then a way like this to associate fields:
https://gitlab.com/tikiwiki/tiki/-/merge_requests/564 (see image)

Tiki Manager tracker field type
Let's add something like https://doc.tiki.org/Show.t.o-Tracker-Field with generic options to launch actions on various Tiki instances. We could reuse some of the parameters from the plugin. Ex. to restrict to only backups and updates.

dev/staging/prod context
The workflow is context to the organization and project. Some users may be allowed to push from dev to staging, but not staging to production. More thought will be need on this aspect, which we can focus on once higher priority use cases are handled.

- Sync Dev-Prod Servers
- Configuration Management for Tiki Projects
- Configuration Management and Systems Orchestration
- Divergent Preferences in Staging Development Production

Things to add
There are many things to add to address appropriately all the main use cases.

- Create or clone instance using
  - tag/branch
  - merge request
  - cherry-pick
  - profile
  - etc.
- Site should not start with admin/admin
- KPIs (already available via Tiki Manager) should be made available for reports and automation.
- Login as admin via Tiki Manager. Similar to Remote Auto-login but perhaps done with Oauth?
- A way to set more aspects of a site
  - How long site will be open. Ex.: for demos, 90 days
  - Backups handling. Ex.: no backups, via Syncthing, etc.
  - Monitoring. Ex.: no monitoring, Virtualmin, etc.
  - Many more will be added over time...

Questions
- How does instance list WYSIWYCA work?
  - A new field for the owner of the instance? (Should be a multiselect username to be future-proof)
  - If someone tries to create or manage a Tiki instance owned by someone else: it should fail.
- Do we (and if so how) limit the number of created sites?
- We could build rules like this with PluginList and Calculations
- How would we do forms like this? https://tikitrackers.org/Demo
  - We split in two:
    - To have your own demo instance for 90 days, you register an account and click to add.
    - To use common demos (which can be busted): it stays like now (but we add a daily reset)
- How can we do mass actions?
  - From list of all Tiki instances, which connect via various instances of Tiki Manager
  - Apply a profile, apply a merge request, make a backup, etc.
  - Could we extend PluginListExecute?
    - So we could configure to make a clone of template.casemanagement.demo.evoludata.com to casemanagement.demo.evoludata.com after no one has used for 1 hour (which we'll know from the KPIs)

**Example flow for a Tiki demo creation setup**

Tiki Manager -> DNS has been set for *.lab1.wikisuite.org and lab1.wikisuite.org

Add a new Tiki instance

Connection Type (Tiki Manager has previously been connected to some Virtualmin instances, and thus, offers us options)

- local
- SSH
- FTP
- Virtualmin dev1.wikisuite.org
- Virtualmin lab1 <- **We choose this one**
  - manager.abc.lab1.wikisuite.org
  - manager.def.lab1.wikisuite.org
  - manager.xyz.lab1.wikisuite.org <- **We choose this one**
- Virtualmin dogfood1

System requests:

- Email
- Password (if left blank, it will be sent by email)
- Instance/domain name
  - Default pattern is alpha.xyz.lab1.wikisuite.org
- Version + profile

Result:

- System creates a sub-server (later, we can add more options, like use Sub-Server, or create/use Top-level server)
- System creates Tiki instance
- Send email to confirm Tiki is ready, with URL and username (If no password was set, use random password and send in email)
Related

- Manager
- Tiki and VirtualMin interop
- Tiki Manager Packaging
- Tikis remote management tools
- https://wikisuite.org/PHP-code-for-VirtualMin

Alias names for this page:
TikiManager in Tiki | Tiki Manager in Tiki | TikiManager within Tiki