



Business Cyber Rigging

----- (---)

Purpose:	Deploy a range of open-source programs to amplify —'s operations, and reduce dependency on corporate programs.
Proposed by:	Operand Company
Lead:	C--- Y---
Referred by:	----

Experience

After around six years of independent cyber research and experimentation, Operand Company has practiced many methods of de-corporatizing online communications.

From running our own email in a home lab made of raspberry pi circuit boards, to deploying a virtual-machine farm with password managers, graphic design suites, home-made blogging engines, and organization group chats, the complexity of “escaping the cloud” has become a daily practice that is ready to be shared more publicly.

This process should become commonplace and commodified, so a focus during this commission and deployment is going to be on producing repeatable guidance; **a guidebook and upkeep manual.**

This manual shall describe the process for launching and managing organizational programs, rigging up users and access management, and harnessing modern security practices to keep the programs hidden from possible attack. The manual is going to be a public record of our process to deploy programs for —'s business use.

Sequence

1. Deploy a shared machine, and share access credentials. For simplicity, this can be a cloud virtual machine through DigitalOcean or Linode. A physical computer is also possible, accompanied by some additional labor demands.
2. Use [NixOS Anywhere](#) to configure [Nix OS](#) on the shared machine.
3. Deploy copies of [Gitea](#) and [Wiki-js](#) on the shared machine; this can be done immediately using NixOS.
4. Add a group of users into these programs. Begin filling in the private codebases and documentation pages necessary so far, keeping access credentials off of the documentation.
5. Deploy a copy of [Passbolt](#) (or possibly [BitWarden](#)) as an organization password manager. Add users and upload the shared credentials for managers. This is a more nuanced deployment, depending on containers managed through either [Podman](#) or [Incus](#). Because this program is more crucial to secure, additional guidance needs to describe backup, migration, upgrade, and account-recovery procedures.
6. Choose additional programs to amplify collaboration. Each additional program becomes a shared resource to enable future operations, deployments, and changes.

Program Schedule

Operand keeps an eye on compelling open-source programs for business use. You can explore [the programs on our radar](#). Though some of these are generically useful, others need to be combined and composed to make sense for specific business aims. Many of these programs have guidance already public through an array of YouTube channels dedicated to this niche.

Recommended deployments include:

Phase	Name	Purpose
origin	Linode / DigitalOcean	Managed 24/7 cloud virtual machine, simplifying shared access.
origin	Nix OS	Fully dependable language / operating system for the shared machine.
origin	Gitea	Similar to GitHub , holds a record of group source code.
origin	Wiki-js	Full wiki / knowledge base for the organization to communicate.
base	Podman / Incus	These are commonplace behind-the-scenes “container” programs; they enable a single machine to play the roles of many machines. Containers expand the range of possible deployments, & keep secure program bounds.
base	Passbolt / BitWarden	Open source password & secret management.
base	CryptPad	Secure documents, spreadsheets, presentations.
base	BaseRow	No-code database & application builder.
upgrade	Penpot	Design platform for teams that need scalable collaboration.
upgrade	Ghost	Modern tools to build a website, newsletters, & paid subscriptions.
upgrade	Cal.com	Fully customizable scheduling software for businesses taking calls.

If — decides to add more programs to the schedule, we can probably manage a quick launch; please be encouraged to bring up new ideas!