Andi Leach.

Software engineer, sysadmin, etcetera.

Personal.

Contact Details

Web	//duck.moe
Email	me@duck.moe
Github	/duckfullstop

Pronouns

They / Them
I'm non-binary!

They are a really nice person.
I asked *them* for an interview.

About me.

Whilst I might be quite a quiet, reserved kind of person, that hasn't stopped me from tinkering with everything put in front of me since I was about 5! I'm entirely self-taught, and have built up almost all of my knowledge and experience simply from experimentation and personal enjoyment, as well as from my numerous projects over the years. When I'm not learning about new things, travelling the planet, or working on one of my many projects, you can usually find me either tinkering with electronics projects, geeking out *far* too much about Electric Vehicles and transit infrastructure, on a long walk, or playing Japanese Arcade games!

Interests.

My present main interest lies with building distributed, zero-downtime services on the server side, including with Kubernetes, Microservices, and other associated technologies, as well as in system administration & ops.

Experience.

Lead Developer & Project Manager
@ *Flower*, 2015 - 2018

python, flask, mariadb, php, laravel, go, redis, docker, sentry, graylog, project management, reverse engineering

Flower is a non-profit, independent, international Arcade Network, with a focus on imported Asian Arcade games. Since my co-founding of the project in 2015, I grew the network from small roots with only one arcade, less than 20 players, and 3 co-founders, to supporting over 15 locations across Europe, over 650 players, and a worldwide development team of over 20 volunteers from a diverse set of backgrounds. The software suite we worked on is still considered the gold standard in its field, and is used by numerous others across the world.

To give just a sample of some of the things I accomplished at Flower:
- Oversaw the direction of the project from inception, ensuring that constituent arcades and players remained happy and informed of the project's progress, and that our development team's members were confident in their work and had the opportunity to fully express their creativity
- Developed the majority of our arcade communication server, with a Python 3.6 / Flask / SQLAlchemy (MariaDB) based stack *from scratch*, based on reverse engineered data from other team members - this included architectural decision making from day 0 and throughout
- Developed the majority of our Web Interface, written in PHP / Laravel 5, including substantial experience with the Bootstrap framework, SASS preprocessing, and jquery
- Worked on other tooling as required, including a C++ game metadata processor, and later some Go microservices for real-time, event-based notifications & postprocessing
- Made extensive use of Version Control (git) to manage numerous branches and forks of the above applications, including handling some pretty heavy rebase conflicts at times!
- Configured and managed the service infrastructure on Azure with Debian / Docker deployments, including an active Gitlab server, automatic CI and CD with gitlab-ci, and Sentry & Graylog for rapid response to incidents
- Designed logos, posters, and other graphical assets as required

Production / AV Lead
@ *theBIGGAME*, 2014 - 2018

rtmp, node.js, nodecg, adobe suite, (live) video production, event management

theBIGGAME is a long running, quarterly LAN Party (gathering of PC Gamers) in the Southampton area. With over 40 events in the past 20 years, it has grown from very small beginnings to a gathering of over 100 participants, continuing to regularly sell-out within weeks of availability and attracting sponsorship from brands including Novatech, Dominos, Plantronics, Tesoro, and i-rocks, among others.

Some of the things I was responsible for included:
- Managing the event's twitch.tv live stream, which included setting up IP cameras, HDMI, and NDI feeds to a short pre-event deadline, producing and managing these feeds using the vMix live production switching software throughout the event, and moderating the community chat
- Producing fresh and current graphical assets in Photoshop, Premiere, and InDesign for use both as Stream content and for large scale print
- Developing dynamic, animated, and vibrant HTML / Javascript / node.js for use with our nodeCG Graphics suite, which is used to power the event's dynamic graphics
  - these graphics received data from a variety of sources, including the last.fm API, Google Calendar, and an internal dashboard, helping us cut down on manual entry
- Delivering new, inventive solutions to problems within a tight budget, including Raspberry Pi-based Remote IP Cameras, and a custom nginx-rtmp based Internal Streaming solution to reduce bandwidth and improve reliability
- Managing talent and assisting guests to ensure the smooth running of the event, including commentary teams

Open Source & Community.

I've created and contributed to a number of open-source and community projects over the years.
To give just a brief taste:

- Handled responsible vulnerability disclosure (including a proof-of-concept written in Go) to a major vendor
  - **Presently under live NDA**, further details when disclosure complete
- Wrote *gopenttd*, a game server communication library, in asynchronous Go
  - *github.com/ropenttd/gopenttd*
- Wrote helper utilities for OpenTTD in containers, *docker_openttd* and *openttd_k8s_helpers*
  - *docker_openttd* packages OpenTTD into super-lean OCI-compliant containers with full Github Actions CI, and includes *Kustomize manifests for Kubernetes*
    - *github.com/ropenttd/docker_openttd*
  - *openttd_k8s_helpers* provides helper runtime utilities to help make the above container more orchestration-friendly
    - *github.com/ropenttd/openttd_k8s_helpers*
- Administered and moderated one of the most popular *OpenTTD* game servers on the server browser, running on a NixOS / *Kubernetes Bare-Metal* stack
- Made codebase contributions to projects including *EVCC, nixpkgs, Longhorn,* and more
- Publish my homelab's declarative infrastructure on Github
  - My personal infrastructure presently runs on a mix of NixOS and Kubernetes, with Kubernetes initialised *from bare components* as an IPv6/IPv4 dual-stack, mixed-architecture cluster. Monitoring and alerting is handled with Prometheus, Grafana, Jaeger, and Loki, while continuous deployment is achieved with Flux.
  - *github.com/duckfullstop/infra*

References.   **Available on request.**