SRE for Smaller Organisations

Applying larger scale to the smaller

Allan Shone, Senior Systems Engineer at Deputy.

@CerealBoy

What is SRE?

Site Reliability Engineering

- An alternative approach to the DevOps methodology as defined by Google.
- Focuses heavily on the automation aspects of removing Toil, but with data to back the decision making processes.
- Begins with system visibility, data gathering and crunching, culminates in action with the overall goal to be reliability in general.

The Problems to be Solved

What is the reason for having SRE any way?

- Toil helps with rapid growth but can easily become unsustainable.
- People burnout from repetitive / menial tasks.
- Decisions shouldn’t be made on a whim or a hunch.
- To be more autonomous people require the right capabilities to manage themselves.

Within a Smaller Organisation

What this means for a smaller organisation, where the freedoms required may not be amenable to SRE

- Likely, SREs will be Ops, Sysadmins, Devs, and Support.
- A lot of the SRE (from the book) components will become Nice To Haves™.
- Focus needs to be applied in the right areas at the right times.
- SRE may feel very much like an older Sysadmin type setup, but with the development methodologies, though making use of Monitoring as a part of the processes.

The 5 Principles

Base principles to build an SRE structure up from

Everything should be Reproducible

- Infrastructure as Code
- Don’t be Perfect
- Iterate over time
- Toil is inevitable
- Plan from DR
- Document *everything*

As much as necessary should be automated

- Toil is often avoidable
- Remember principle 1, though
- Rare events are unnecessary
- Maintenance is still a factor

Security should not be ignored

- Starting well saves time
- Coding standards simplify things
- System standards also help
- Processes for system management go a long way

Simple is better than Magical

- Naming things appropriately
- Time taken to explain systems is better spent developing
- Explicitness grants transparency
- Assumptions are bad

Visibility is key, Access is not

- Monitoring / Metrics / Logs
- Data extraction
- Read only infrastructure
- Applications self-managed
- Cattle, not pets

Bringing them together

- An alternate Ops (or DevOps) perspective to use.
- More focused on execution of systems and their availability as a measure for success / health.
- Plenty of room for massaging into organisations based on desire and capability.
- Huge amounts of expertise involved.

Thank you!

@CerealBoy