Google Summer of Code 2020 results
Samba AD DC Cockpit UI

Alexander Bokovoy
Samba Team, Red Hat
A starting note

• Google Summer of Code: Samba edition
• Goals
• Results and demo
• Future
A starting note

- Google Summer of Code: Samba edition
- Goals
- Results and demo
- Future

- About me
  - Samba Team member
  - Engineer at Red Hat
  - Focused on identity management and interoperability
• Projects and resources
Google Summer of Code 2020

- Projects and resources
- 4 proposals, 1 seat
• Projects and resources
• 4 proposals, 1 seat
  • Wireshark dissectors for Print System Asynchronous Remote Protocol (MS-PAR)
Projects and resources
• 4 proposals, 1 seat
  • Wireshark dissectors for Print System Asynchronous Remote Protocol (MS-PAR)
  • Cockpit UI prototype to manage Samba AD DC
Google Summer of Code 2020

• Projects and resources
• 4 proposals, 1 seat
  • Wireshark dissectors for Print System Asynchronous Remote Protocol (MS-PAR)
  • Cockpit UI prototype to manage Samba AD DC
• the other two proposals weren’t up to expected level
• Projects and resources
• 4 proposals, 1 seat
  • Wireshark dissectors for Print System Asynchronous Remote Protocol (MS-PAR)
  • Cockpit UI prototype to manage Samba AD DC
• the other two proposals weren’t up to expected level
• we decided to go with a Cockpit UI proposal
Google Summer of Code 2020

- Projects and resources
- 4 proposals, 1 seat
  - Wireshark dissectors for Print System Asynchronous Remote Protocol (MS-PAR)
  - Cockpit UI prototype to manage Samba AD DC
- the other two proposals weren’t up to expected level
- we decided to go with a Cockpit UI proposal
  - GSoC is an effort for both a student and mentors
What is Cockpit?

Introducing Cockpit

Cockpit is a web-based graphical interface for servers, intended for everyone, especially those who are:

- new to Linux (including Windows admins)
- familiar with Linux and want an easy, graphical way to administer servers
- expert admins who mainly use other tools but want an overview on individual systems

Thanks to Cockpit intentionally using system APIs and commands, a whole team of admins can manage a system in the way they prefer, including the command line and utilities right alongside Cockpit.

Take a look

A picture is worth a thousand words. Click a thumbnail to see screenshots of Cockpit in action.
GSoC student

• Hezekiah Maina
GSoC student

• Hezekiah Maina
  • University of Nairobi, Kenia
• Hezekiah Maina
  • University of Nairobi, Kenya
  • Bachelor’s degree, Real Estate 2017-2021
Hezekiah Maina
- University of Nairobi, Kenya
- Bachelor’s degree, Real Estate 2017-2021
- Experienced with JavaScript and React
GSoC student

• Hezekiah Maina
  • University of Nairobi, Kenia
  • Bachelor’s degree, Real Estate 2017-2021
  • Experienced with JavaScript and React
  • New to Linux, new to Samba
The GSoC program has several goals:

• Inspire young developers to begin participating in open source development
• Help open source projects identify and bring in new developers
• Get more open source code written and released for the benefit of all
• Provide students the opportunity to do work related to their academic pursuits during the summer: “flip bits, not burgers.”
• Give students more exposure to real-world software development (for example, distributed development and version control, software licensing issues, testing, and communication best practices)
Samba GSoC goals

- Understand how can we bring “Samba world” and “Web world” together
- Explore open source communities ways of working
- Improve over a previous prototype done in 2018
- Learn how to do continuous integration and delivery for Linux
From my previous talk at sambaXP 2018
How complex is it to manage Samba?

• Five main server roles:
  • Standalone server
  • Domain member server
  • Classic primary domain controller
  • Classic backup domain controller
  • Active Directory domain controller

• File share configuration
  • Applies to all five roles
  • allows 133 different options per share

• Global configuration
  • 339 different options
That was just `smb.conf` configuration

- Databases beyond `smb.conf`
  - identity information backend
  - secrets database
  - account policy database
  - SMB identity to POSIX group mapping
  - NetBIOS browsing details database
  - Kerberos keytabs

- Utilites
  - `net`
  - `samba-tool`
  - `smbcontrol` (an instant messaging app)
  - ...
Cockpit concepts

- Use of system APIs and commands you already can use from the shell
Cockpit concepts

- Use of system APIs and commands you already can use from the shell
- Present information already available in the system
Cockpit concepts

• Use of system APIs and commands you already can use from the shell
• Present information already available in the system
• Make Linux discoverable, don’t add additional layers
Samba AD

- Samba AD primary tool is `samba-tool`
- Manual steps needed for deployment
What was that?

- Cockpit management console for Fedora 35 (Rawhide)
- Samba AD Cockpit application
  - prototype based on the GSoC results
  - deploys Samba AD domain controller
  - manages Samba AD domain
  - or shows its state
- Behind the (web) interface
  - Runs `samba-tool` commands
  - Parses output
  - Presents the results in UI
What was that?

• Cockpit management console for Fedora 35 (Rawhide)
• Samba AD Cockpit application
  • prototype based on the GSoC results
  • deploys Samba AD domain controller
  • manages Samba AD domain
  • or shows its state

• Behind the (web) interface
  • Runs `samba-tool` commands
  • Parses output
  • Presents the results in UI
Cockpit application

• React-based JavaScript application
• Uses existing Cockpit APIs to integrate in the UI
• Part of Cockpit app, socket-activated and authenticated
• Cockpit session is like an SSH session
  • Properly authenticated, can use sudo, if required
  • All you can do in SSH session can be done by a Cockpit app
    • even to a remote Cockpit server
Cockpit Samba AD UI

- Result of GSoC: https://wiki.samba.org/index.php/GSOC_cockpit_samba_ad_dc
GSoC challenges

- Continuous CI
  - GitLab to host the project
  - OSBS to create automated builds (RPM and DEB)
- Automating NPM builds without internet access
- Documentation hurdles
- Parsing free text output from `samba-tool`
- Development against moving targets
Challenges (part 2)

• How to productise GSoC results?
• Keep in sync with Samba development
• Produce playbooks for common tasks
• Samba is used by people
• And robots
• Robots increasingly consume Samba artefacts
• Parsing human-oriented output is a waste of resources for robots
• We can do better (for robots and humans)
• A little magic can help both
Thanks!