Proposal for infrastructure Working Group

Summary

1) Set up a working group for the infrastructure.
2) Hire a contractor to do the work.

Author: Christian Kühnel
Contributors:
Intended audience: LLVM board, community & ops team

Status: Draft
Created: 2020-09-16

Self Link:
Context: LLVM
Visibility: Public

Intention of this document

This document is meant as an actionable proposal on improving the LLVM infrastructure. It is explicitly intended to start and encourage a discussion in the LLVM foundation and community on organizing infrastructure work in the future.

Contents

Problem statement

The LLVM project runs several infrastructure services to support the community in developing software. This infrastructure - e.g. Github, Phabricator, BuildBots, Bugzilla, Mailman - is crucial to keep the community alive and be able to work efficiently and effectively on a project of this scale. The infrastructure is maintained and improved by community members. Many of these community members are paid for this work by their respective employers[1]. There is also previous experience in hiring a contractor for migrating the BuildBots.

There is no central body in the project to drive and decide on the infrastructure work, it's rather a group of interested, trusted people proposing and implementing infrastructure changes. There is also no clear way to decide what the agreed infrastructure and workflows should be in the future and how to get there. However, in many places the community needs to decide on exactly one piece of infrastructure they want to use or one way of doing something. Chris Lattner recently introduced a decision making processes, however this has not been used so far for infrastructure changes.

This has some drawbacks:

- There is no common vision for the project infrastructure or the intended workflows. There are no design principles that can guide decisions on infrastructure changes.
- The project infrastructure is maintained on a best-effort basis. There are no service level agreements in place.
- For some areas, there are multiple things in place with no clear picture on which of those are mandatory. Looking at build servers for example we have the build bots, green dragon, Github Action, Buildkite and pre-merge testing. All provide value, but do not give a consistent picture. Coming to a clear picture is not only a technical but also a political task.
- Progress on infrastructure improvements are quite slow. There are some hypothesis why that is the case:
  - Hypothesis 1: It's very difficult to come to a decision on disruptive infrastructure changes, as any change will be opposed (loudly) by some person on the mailing list. Especially if a change impacts people’s workflows or company workflows downstream.
  - Hypothesis 2: Currently (though improving) our infrastructure is very fragile and so those of us working to improve things tread very lightly as we have all had the experience of losing entire weekends "trying to make things better".
  - Hypothesis 3: Even if folks have their employer’s support for working on these projects, they are of low priority. And once something more pressing comes along, infrastructure work has to be deferred.
- Compiler experts are building and maintaining the project infrastructure and servers. They are not experts for this type of work. The compiler developers see their work mostly in doing compiler work, this is where they draw their work satisfaction from. Thus the infrastructure always has a lower priority for them. Experts on compilers are hard to find on the market. Spending their time on infrastructure work is not efficient.

[1] Looking at the commit history and the TOP 20 email domains, 66% are from corporate accounts, for 33% it’s unclear (e.g. @gmail.com). The TOP 20 email domains make up for about 60% of all committers.

- Companies are donating the time of their employees to maintain the infrastructure. But for larger companies it might be preferable to donate money instead of headcount for the maintenance of the infrastructure. In such companies, headcount is usually harder to come by than money.

Proposal

To overcome the challenges, this document will propose an approach to run the infrastructure of the LLVM project. The proposed solution will consist of two parts:

1. Transition the current ops team into an “infrastructure working group” to decide and drive improvements in the infrastructure and
2. pay someone to do the improvement and maintenance work for the infrastructure where needed.

The infrastructure working group (short: working group) shall investigate the current project infrastructure and understand the needs and pains for the community. The working group will propose the future direction for the infrastructure guided by a common vision. The working group will propose work items based on that vision to the LLVM board who will then approve these work items. The work items shall be executed by a (potentially paid) community member or contractor. The infrastructure team will work on a day-to-day basis with them to set up, maintain and improve the project's infrastructure.

The working group shall define the workflows the infrastructure shall support, based on the current situation and the input from the community.

The working group shall be judged by their impact on:

- stability of the infrastructure
- efficiency and effectiveness of the work flows and tools
- happiness of the community (e.g. based on surveys)

The next sections provide more details on the proposal. All of this is just an initial draft to get things started and shall be re-visited and re-negotiated as needed.

Definition of infrastructure

The infrastructure of the LLVM project includes:

- source code repository
- bug tracker
- all build servers/machines
- all web sites
- build system/scripts
- code review tool
- mailing lists
- internet forum
- chat (IRC and discord)
- Documentation of the tools and their work flows.

All of these systems shall be managed/owned/administered/coordinated by the working group. They recommend to the board which of these shall be handed over to a contractor for maintenance. The working group shall also define service level agreements (SLA) for all of these tools and the intended workflows. The respective contractor shall guarantee this service level for community members. The working group can propose to the board to change, replace or add tools, and workflows. Changes with significant impact on the workflow of the developers need to get approved by the community.

Setting up a working group

The first step in the process is to establish an infrastructure working group by either creating a new group or changing the work mode of the existing “ops team”. The working group shall be composed of around 5 community members and 2 board members. One of them shall be a designated chair for the working group. These community members shall be proposed by the community and appointed by the LLVM board. The chair shall be nominated by the working group and confirmed by the board.

The **working group members** must be willing to spend around 4 hours per week on project management work (writing/reviewing work packages, engaging in community discussion, guiding the contractor, defining roadmaps, ...). In addition the meeting group members shall meet once a quarter on a video conference and if possible meet twice a year in person (Euro LLVM conference and US LLVM dev meeting) to discuss the major topics for the next 6-12 months. The working group shall also present the work of the previous 6 months and the outlook for the next 6 months during these conferences to the community.

The **chair of the working group** shall host these meetings and guide the work of the working group. They are also the contact person for the board and shall coordinate the work with the board. The chair is expected to dedicate around 8h per week for this job. The chair shall also be involved in the budgeting process of the foundation to ensure funding of planned infrastructure work and maintenance.

All discussions and decisions of the working group are open and transparent to the community. Everyone in the community is encouraged to participate in these discussions. Major changes in the infrastructure will be run by the community using Chris Lattner’s new decision making process. Sensitive issues related to contracting (e.g. contracts, offers, prices, ...) are not discussed in public. The discussions and decisions shall be done by the working group as much as possible in a written, asynchronous way, so that the working group and community members are free to plan their work as is appropriate for them.

Decisions in the working group shall be taken by a simple majority.

The reason for the transparency towards the community is to strengthen the trust of the community in the working group and avoid conspiracy theories.

Working with the contractors

The working group shall create a statement of work (SOW) for a planned change. They shall then decide on how this shall be implemented, depending on the type of work and the required service levels. The usual options would be:

- There is a volunteer or a group of volunteers to implement the work.
- The working group offers a bounty to a community member for the implementation.
- The working group runs a request for proposal with a group of contractors.

The working group will propose the SOW to the board for approval and contract negotiations. While the work of creating SOWs and contracts will require time and effort from the working group, there is significant leverage in offloading the technical work to someone else. Especially the constant maintenance of a service with a SLA is a significant long term effort and shall be offloaded to someone who has the expertise and time to do that.

A typical SOW will contain an overview of the workmode, the required skills, the functional and non-functional requirements, the work items, the acceptance criteria, and the service level objectives.

All contractors shall maintain a public documentation of the systems and SOWs they are responsible for. The working group will also maintain a method of being able to cut ties with a contractor instantly without disruption to the community. This will be a non-negotiable point as part of the contracts.

Paid community work

Another option of getting work done is to offer bounties for completed SOWs to community members. Especially for one-time changes, this is a nice way of giving something back to the community. The work mode will be the same as for the contractor: The working group will define the SOW and afterwards review and approve the results.

Day-to-day work

This section describes flow for the day to day work of the working group. Note that only the good case is described here. A proposal can be cancelled in any of these steps. There might also be some iterations as some parts of the SOW are not clear or feasible.

1. Anyone in the LLVM community (including contractors) can propose a change.
2. The working group will create an SOW based on the proposals they want to move forward on. The SOW will consist of a problem description and (functional and non-functional) requirements for a solution of the problem. The SOW shall be made public.
3. Once the RFP and it's costs are clear enough, the working group will decide if it shall be implemented and then propose it to the board.
4. The board will then review and approve the expenditure.
5. The Foundation will set up the contract and manage the payments.
6. The working group will work with the contractor on a day-to-day basis to get the work done.
7. The working group will approve the work and authorize payments with the treasurer.

All of this shall be open and transparent for the community. Feedback from the community is appreciated in all of these steps.

In parallel to this, community members are welcome to take over certain SOWs and will get help from the respective contractor if needed. If that work touches shared infrastructure or impacts the future maintenance, the solution needs to be agreed with the working group and the contractor.

Benefits and drawbacks

The next sections will list the benefits and drawbacks related to the proposal above.

Benefits

- The LLVM community will get a professionally maintained infrastructure with Service Level Agreements (SLA) guaranteed by the contractors. SLAs will make the infrastructure performance more predictable.
- The infrastructure is independent of contributions/experts of a particular company.The compiler experts can focus on compiler-related work.
- The infrastructure work is managed by a small group familiar with the topic. This group can quickly and efficiently reach decisions on small changes.
- The infrastructure is maintained by experts familiar with the topics. They can focus on this topic. Their work motivation comes from efficient workflows, reliable tools and a happy community.
- Some companies can move from dedicating head count to money. This could also free up time for the compiler experts.

Drawbacks

- The LLVM foundation is dependent on a steady flow of donations to be able to pay for the contractor.
- By handing over the infrastructure to a contractor, the community is losing knowledge and is becoming more dependent on outside experts.
- Deploying infrastructure changes quickly might alienate some contributors as they are used to a rather static environment right now and they have settled with its current problems.
- Creating SOWs and contracts will require additional effort and people willing to do more of a project management type of work.

Appendix

Ideas for initial work packages

This is an unstructured list of ideas for initial work packages for the working group and the contractors:

- Create and maintain a public documentation of the existing infrastructure and the supported workflows (what, why, who).
- Define and agree a vision for LLVM infrastructure and for the design principles thereof.
- Identify the need for addons for Github to implement features that the community is missing right now.
- Come to a decision on using GitHub Pull Requests, then implement the change.
- Create a roadmap for the handover of the existing infrastructure to the contractor. Components ready for immediate takeover:

    - Phabricator (in case we stay with it)
    - Pre-merge testing
    - Buildbot workers
    - Remote Index for LLVM
    - ...
- Create new design and structure for the LLVM website, migrate the existing content there.
- Create a test strategy for the LLVM project and refactor the existing build infrastructure to meet the new test strategy. Some ideas:
    - Multi-stage CI: run “cheap” builds first, run “expensive” builds only on commits where the cheap builds have passed.
    - Upgrade buildbots to a current version, also deprecating Python2 along the way.
    - Set up a collector dashboard showing build results for all CI systems per commit, e.g. Treeherder or something GitHub based.
    - Investigate if we can get a good deal from GitHub Actions with sponsored hardware.
- Gather user feedback on the existing infrastructure: Where are the pain points? What would our community like to see improved?
- Migration of all python scripts and tools to Python 3 as Python 2 is deprecated.
- Reach out to downstream infrastructure teams (e.g. Rust, Tensorflow) and understand if we have infrastructure topics we want to cooperate on.
- Define Service Level Objectives (SLO) for our infrastructure and monitor them. Set up monitoring and altering for our infrastructure. Derive measure where we do not meet our SLOs.
- Migrate Windows setup instructions off GnuWin32 as it’s deprecated and add installation instructions users can copy-and-paste into their shell.
- Investigate the Rust project infrastructure and see what would be helpful to LLVM. Some examples:
    - Create a benchmark tool to compare the performance of the compiler and the compiled code across versions, there is also a project for that.
    - Set up some central source of truth for managing user accounts, group membership and user/group permissions. Then configure all tools accordingly.

Elevator pitch for working group members

This is intended as an email sent to the LLVM community and folks potentially interested in joining this working group.

> Hi folks,
>
> **tl;dr**
> We’re looking for community members interested in joining the new “Infrastructure Working Group” to shape the LLVM development infrastructure.

# Background
The LLVM project infrastructure has grown over the years and progress has significantly slowed down. Our aging infrastructure, combined with the growth of the community means we need help upgrading the tools and automation relied upon by our every growing community.

# Proposed solution
The LLVM foundation is in the process of launching a new “Infrastructure Working Group” (IWG) composed of community members. The goal of this working group is to improve the development infrastructure of the LLVM project.

The focus of the IWG will be on identifying, designing and agreeing on infrastructure improvements. The implementation of the changes and the operations of our infrastructure can be offloaded to community members or contractors. This way we keep the workload for the working group low.

We would like to have broad and diverse community representation within the working group as we are convinced such groups deliver the best results.

# Call for participation
If you’re interested in joining the working group, please reach to TODO

Also please forward this email to folks who might be interested in joining!

Best,
Christian Kühnel
Tanya Lattner
Mike Edwards

TODO list for startup of the working group

- Finalize the invitation email (see above) and initial web site.
- Get email account set up
- Gather a group of initial members and work out appointing them to the committee
- Hold an initial meeting of the Group
- Set some initial goals and milestones for 2021
    - Run survey to understand biggest issues for contributors
    - Document current infrastructure
- Prepare a quick report to provide the Board at the February meeting
-