Question 1

If you are going to install something on Linux, what are the different ways that could be done?

If there is a system package (e.g. rpm,deb,etc) I'll install using the appropriate manager (dnf/rpm,apt-get/dpkg,etc) as ideally most if not all dependencies would installed automatically.

If a package isn't available, I'd compile from source (either a git clone or release tarball). Typically instructions for compiling are laid out in the README. This usually involves:

- install necessary dependencies
- executing a config script to define variables used during installation and to ensure dependent libraries exist. Typically I'll set `--PREFIX=/usr` for system consistency (personal preference).
- utilizing `make`, `cmake` or another build utility that will compile then place resulting build files where they need to be.

If I'm installing/compiling something in a directory not in my default path (e.g. /opt/sonic-pi) then I will update my .bash_profile to include /opt/sonic-pi/bin in my executable path.

Question 2

What are the advantages and disadvantages of using a Web framework such as Rails?

A huge advantage is the ability to implement a complex design practice without having to start from scratch. And with custom scaffolding templates, setting up basic components (model,view and controller files) can be automated which helps make adding a resource much easier.

Database security is less of a worry with sanitized controller methods.

Also, the undo method in a Rails data migration is quite useful when debugging issues.

One disadvantage is that an entire framework can be a bit much for smaller projects (single page sites or apps that share only one host). A decent amount of resources are needed for optimal performance. In terms of troubleshooting, Rails (or any other framework) present the same challenges.

Question 3

What would you do if you found a bug in your software and wanted to use git to help identify when this problem started?

Using clues from the stacktrace, I'd search git (`git log --stat` or via webui) to see if there have been any recent commits on the file(s) being referenced. Then I would cross-reference the dates from those commits with any system logs (rails,docker,jourald,syslog,etc).

If a log scan didn't pinpoint a time, I'd utilize `git bisect` to search commits between the current bad commit and the last known good commit, compiling each commit found until the bug was reproduced.

Question 4

Describe the resources or steps you may use to resolve a customer’s problem when you don’t know the answer. (Make assumptions wherever you need to about how the GitLab team works).

I'd let the customer know we're working on an answer and that I'll ping them as soon possible. I'll do a bit of research using google/stackoverflow/github as well as referring to any internal knowledge base available. If necessary, I'd ask the customer some follow up questions and perhaps ask them to send along some logs. If time permitted, I may fire up a VM in an attempt to replicate the issue. If the answer still wasn't apparent, type up a quick review with what was found so far, then query the team chat room to see if anyone is available to assist. While waiting for a reply, continuing recon and testing.

Question 5

If you have them, please provide links to what you consider some of your best answers on forums such as Stack Overflow, Reddit, or in projects on GitLab, GitHub, etc.

I don't do much posting these days as I do most of my answering on IRC. If you'd like to send back a couple of example questions I would be expected to answer as a Support Engineer, I'd be more than happy to answer.