dards and style guidelines

and sub-projects. The majority of their backend code is written in Ruby ⬓ and Go ⬓.
for automation of routine system administration tasks like deployment, installation, etc.
ns or as an effort to minimize the dependencies, for instance, for Docker images.

ur shell scripting guidelines, based on our various experiences. All shell scripts across
nonized with this guide. If there are any per-project deviations from this guide, they
`PROCESS.md` file for such a project.

end staying away from shell scripts as much as possible. A language like Ruby or Python
es that we leverage) is almost always a better choice. The high-level interpreted
ffer much more mature capabilities for unit-testing, linting, and error reporting.

estriction on project's dependencies size or any other requirements that are more

irements, this guide covers only those shells that are used by supported Linux

lencies list, use what's provided by the environment. For example, for Docker images it's
age for most of our tool images.

. It's more powerful than `sh` but still a widespread shell.

uld be made a mandatory part of a project's CI pipeline if it contains shell scripts. These
ecking for errors or vulnerabilities, etc.

s default configuration to lint our shell scripts.

this GitLab CI/CD job:

e:stable