Set up Basic GRAIL Developer Environment

Set up Basic GRAIL Developer Environment

In this document, learn how to clone from and prepare to push code to GRAIL code repositories. Install Arcanist, Bazel, Git to contribute to the GRAIL development process.
Be sure to join the sw-announcement Google group as well.

- Install required software
- Add SSH Public key to Phabricator
- Clone GRAIL code repositories
- Prepare to push to GRAIL code repositories

To complete the setup properly, follow the instructions in each section.
Ensure that you complete each section before moving on to the next one:

Install Required Software
Install Arcanist
Generate an SSH key pair
Setup Phabricator
Update Git user information
Clone the GRAIL mono repo
Prepare to push to GRAIL code repositories

> **NOTE:** If you are not onsite and connected to the GRAIL intranet, you will need to connect to the VPN in order to access some systems, such as staging servers or the code repository.

Install Required Software

1. Open a terminal window and install Homebrew:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

Click through any prompts that may appear.

2. Change Homebrew directory ownership:

```
sudo chown -h -R $(whoami) /usr/local/*
```

> **NOTE:** You will be prompted for your device password

3. Use Homebrew to install additional required software packages:

```
brew install coreutils git git-lfs python@3.7 awscli terraform gawk gnu-sed wget curl bazel jq
```

> **NOTE:** For development in any language at GRAIL, please refer to the language specific guide for installing the toolchain that will work with our codebase. There may be additional steps required.

Install Arcanist

Arcanist provides:

- Command-line access to many Phabricator tools like Differential, Files, and Paste.
- Integrates with static analysis ("lint") and unit tests.
- Manages common workflows like getting changes into Differential for review.

For more information see https://secure.phabricator.com/book/phabricator/article/arcanist/

1. Create a directory named *tools* to download Arcanist:

```
mkdir ~/tools
cd ~/tools
git clone https://github.com/phacility/arcanist.git
```

These tools are required to push code to GRAIL code repositories.

2. Update your path to include Arcanist:

```
echo 'export PATH="$PATH:$HOME/tools/arcanist/bin/"' >> ~/.bashrc
echo 'export PATH="$PATH:$HOME/tools/arcanist/bin/"' >> ~/.zshrc
```

> **NOTE:** A previous version of these instructions used `.bash_profile` and `.zprofile` instead of the current files.

Generate a SSH key pair

If you already have an SSH key pair,
continue on to the **Phabricator Setup** section.
If you don not have an existing SSH key pair, follow the instructions below:

1. Run `ssh-keygen` in terminal to generate a key pair.

Respond to each question by pressing **Enter**.
You will be prompted to create a password, this is optional.
Creating a password means you will need to enter it for each connection.
You are not required to enter a password.

Example output from `ssh-keygen`

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/fdetal/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in id_rsa
Your public key has been saved in id_rsa.pub
The key fingerprint is:
SHA256:0000000000000000000X88keJIxBaZtv4zUjFz5r4y0 fdetal@GB0000000
The key's randomart image is:
+---[RSA 3072]----+
|xxo.x*+ .        |
|xx *jx.B .       |
|+x*.xo+ =        |
| x Bx+ +.o       |
|  + o...S.       |
|   .  = B        |
|     o = =       |
|      . E.       |
|       o.o.      |
+----[SHA256]-----+
```

2. A key pair appears in the ~/.ssh folder.

- The private key in `~/.ssh/id_rsa`
- the public key in `~/.ssh/id_rsa.pub`

The same public key may be used on multiple hosts and services;
you can re-use the same public key for other SSH connections.
Setup Phabricator in the next section.

Setup Phabricator

Once you have an SSH key pair,
you are now ready to upload your public key to Phabricator.

1. Go to https://phabricator.grailbio.com/settings/

Click **SSH Public Keys** item on the left (🔧 AUTHENTICATION → SSH Public Keys).

2. Copy public key to your clipboard:

Back in the terminal, run:

```
pbcopy < ~/.ssh/id_rsa.pub
```

> **WARNING:** Public keys end in *.pub*. Do not share your private key.

3. Go back to Phabricator and click ⚙ SSH Key Actions ▾ → ⤒ Upload Public Key.

A window appears with fields for you to name the key paste the contents your clipboard.

4. Create a name for the key; the example key will use the name *id_rsa.pub*.

5. Paste the public from clipboard into the Public Key field: Select the field and press ⌘ + v to paste and click Upload Public Key.

Update Git user information

Configure Git settings to allow you clone and push changes to GRAIL code repositories.
Git was installed earlier in this document using Homebrew.

Edit and use the following commands according to the comments on each line:

```
git config --global user.email "USERNAME@grailbio.com"  # Replace USERNAME with the first part of your grail email
address.
git config --global user.name "FULLNAME" # Replace FULLNAME with your full name.
git config --global pull.rebase true # Configures git to use rebase instead of merge strategy by default when pulling.
```

Example

If your name is Fulano De Tal and email is fdetal@grailbio.com,
you would update your config like this:

```
$ git config --global user.email "fdetal@grailbio.com"
$ git config --global user.name "Fulano De Tal"
$ git config --global pull.rebase true
```

Clone the GRAIL mono repo

> **NOTE:** You will not be able to pull the repo if you're not connected to the VPN.

From your home directory, Run the following commands:

```
cd ~
```

```
GIT_LFS_SKIP_SMUDGE=1 git clone --depth=1 --single-branch \
ssh://git@phabricator.grailbio.com:2222/diffusion/GRAIL/grail.git
```

```
cd grail
```

```
git lfs install --skip-smudge --local
```

Prepare to Push Changes to GRAIL Code Repositories

> **NOTE:** Ensure that your ~/.zshrc is updated to include Arcanist. `echo 'export PATH="$PATH:$HOME/tools/arcanist/bin/"' >> ~/.zshrc`

Set Default Editor for Arc and Git

Arc uses the editor ed by default.
This can be changed to your preferred editor.
Run the following commands but replace *$PATH_TO_EDITOR* with the path your preferred editor.

```
arc set-config editor $PATH_TO_EDITOR
git config --global core.editor $PATH_TO_EDITOR
```

Example

If your editor of choice is Vim installed via Homebrew, you would run the following:

```
$ arc set-config editor /usr/local/Cellar/vim/8.2.0250_1/bin/vim
$ git config --global core.editor /usr/local/Cellar/vim/8.2.0250_1/bin/vim
```

Install Arc Certificate

The first time you run `arc diff` or `arc patch` to push changes to GRAIL code repositories, arc may complain about verifying the centralized Phabricator server. If this occurs, Run `arc install-certificate` according to the onscreen instructions, then run the command again.

Next Steps

Now that you have completed this guide, you are now ready to submit code to GRAIL repositories with Arcanist. In addition, you are incidently ready to use many of the basic engineering resources. After completing this guide, find team specific onboarding information for further instruction:

- Setup Extended GRAIL Dev Environment
- Onboarding checklist for access bioinformaticians, statisticians, data science, machine learning
- Code Management
- using Git and Phabricator
- Go @ GRAIL
- JavaScript/ES6+ Coding Standards

- Infra Team Onboarding