Basecamp Computer Setup - Revised

Created by Jeffrey Burr, last modified on Apr 22, 2020

In your first Basecamp session you will be guided through steps to do the following:

- Create a Pinterest **devapp**, a development server that you use to run the **pinboard** codebase and connect to production services.
- Log in to **Phabricator**, Pinterest's code repository.
- Join Pinterest's **Engineering** project, which gives you access to projects in Phabricator.
- Create your SSH key and store it in Phabricator. This enables you to access the git repositories stored on Phabricator, and to securely SSH into your devapp server and devapp servers owned by others.
- Set up a Docker container for your Pinterest **devserver**, which displays as a collection of tmux windows in your terminal with each addressing a different service.

Basecamp Computer Setup: Pre-Lab Steps

NOTE: Complete these steps the day before your first Basecamp session. Some of the steps require waiting for a response before you can proceed.

1. Join the channel **#base-camp** on **Slack**.

2. Go to https://devapp.pinadmin.com/ and log in with your ldap credentials.
3. Click **Launch > Launch Personal Devapp**.
4. Accept the default settings.
5. Refresh the page.

   Your devapp management panel appears.

   This process to create your devapp will take around 30 minutes.

   You will not be able to SSH into your devapp or get docker running until the process is completed.

   While the system is creating your devapp, under **Pinterdev DNS** you see a loading icon to indicate that your devapp is being set up.

   When the process is complete, the loading icon is replaced by **OK** as shown here:

You can manage the following personal devapp(s):

Instance ID	Launched	Hostname	Release	Size	State	Pinterdev DNS	Action
i-034ec8052e873bbc7	12/18/2019	dev-xixia	Ubuntu Bionic 18.04	c5.9xlarge	running	OK	Select

› Troubleshooting steps for creating your devapp

If you get errors when creating your devapp, try these steps

If you get a message that says that you are unauthorized:

1) Log out and log back in, to refresh your token.

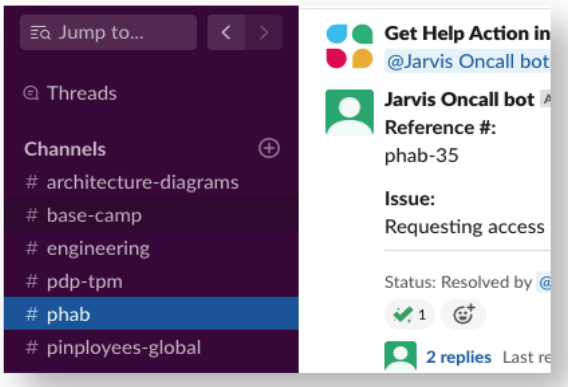
2) If Step 1 doesn't work, clear your cookies, then repeat Step 1.

3) If you are an engineer, email access@ and ask them to add you to the **engineering** LDAP group.

4) If you are not an engineer, email access@ and ask them to add you to the **devapps** LDAP group.

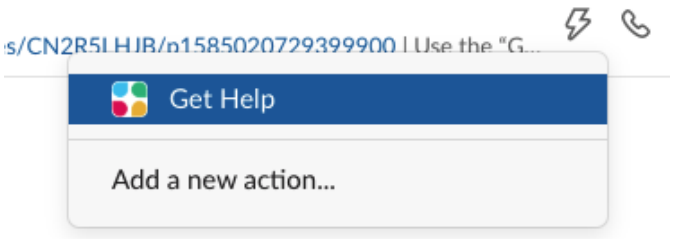
6. Go to https://phabricator.pinadmin.com/ and log in using the Google login.

7. Join the channel **#phab** on **Slack.**

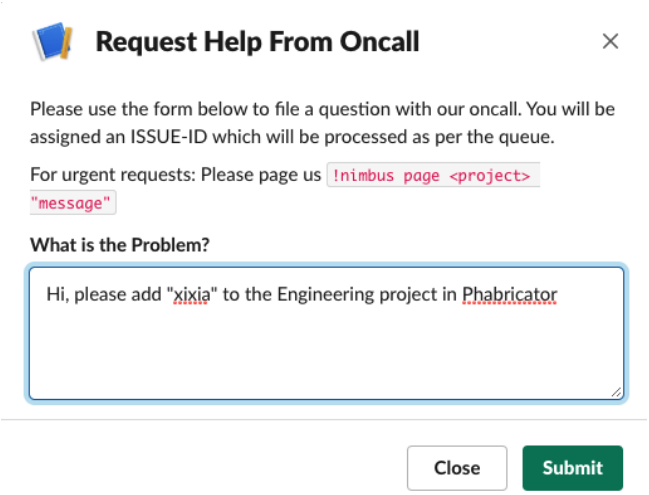


8. In the **#phab** channel, click the lightning bolt icon in the upper right corner.

9. Click **Get Help** from the dropdown menu.



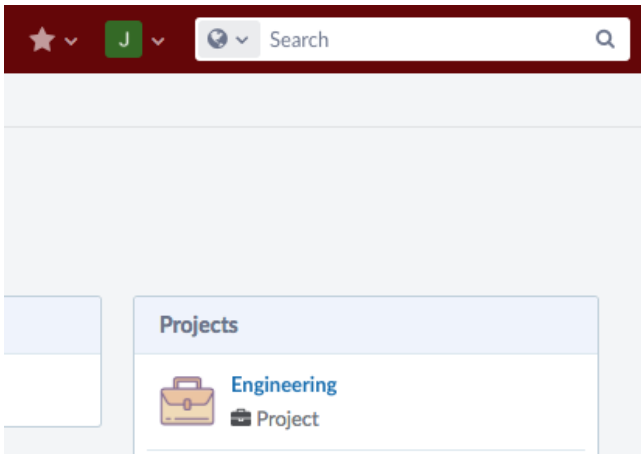
10. The **Jarvis bot** will message you in Slack. Click the **Get Started** button that appears in the message.
11. In the screen that pops up, enter your request to add your username to the engineering project in Phabricator:



The on-call person will confirm when you have been added to the group.

12. Go to https://phabricator.pinadmin.com/ and select **Profile.**

Engineering appears under **Projects** when you have been added to the Engineering group, as shown here:



13. Use ssh-keygen to create a public/private key pair by opening a terminal and at the prompt entering `ssh-keygen -t rsa -b 4096` as shown here:

```
$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/your_username/.ssh/id_rsa):
```

14. Press **enter** to accept the default path.

15. Create a passphrase and save it somewhere on your computer where you can retrieve it. You can add it to your Mac Keychain using the Keychain Access app, as described here. You will need this passphrase again, so don't lose it.
Example: `drumfish henbit dihedral cavern`

16. Enter your passphrase twice as requested. Something similar to the following appears:

```
Enter same passphrase again:
Your identification has been saved in /Users/your_username/.ssh/id_rsa.
Your public key has been saved in /Users/your_username/.ssh/id_rsa.pub.
The key fingerprint is:
72:b5:3f:c9:5b:a4:d5:af:04:e1:d8:32:f7:14:e9:04 your_username@your_computer.local
The key's randomart image is:
+--[ RSA 4096]----+
|            E    |
|             . . |
|          . . +  |
|         . = + o |
|      . S = = = .|
|       o   * O  .|
|            * + .|
|             = . |
|            . .  |
+-----------------+
```

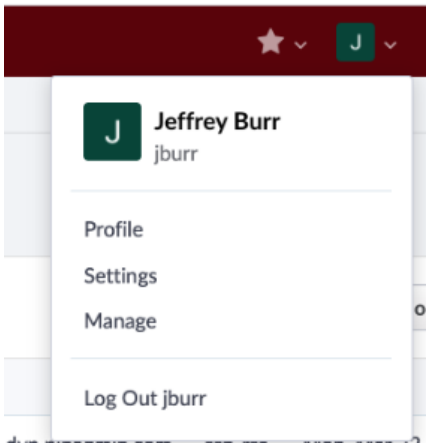
17. Add your key to the SSH agent and OS X keychain by entering this in the terminal:

```
$ ssh-add -K ~/.ssh/id_rsa
```

18. You need the SSH key for the next couple of steps, so copy it using this command:

```
$ cat ~/.ssh/id_rsa.pub | pbcopy
```

19. In your Web browser, go to: https://phabricator.pinadmin.com/settings/panel/ssh/
20. Ensure that you are logged into Phabricator by selecting the profile button next to the search bar. You should see your profile.



21. Click **SSH Key Actions** in the upper right side.
22. From the dropdown list, select **Upload Public Key**.

    The **Upload Public Key** screen opens.
23. Enter a name of your choosing to identify what the public key is for, for example: **my-macbook**
24. Paste your previously-copied SSH key in the **Public Key** text box. This is the key that you copied in Step 18 above; it should be in your clipboard ready to paste.
25. Select **Upload Public Key**.

    NOTE: To complete the following step you must be part of the engineering project in Phabricator. If you have received a response to the request you made in Step 11 to be added, you can proceed.

    If you haven't received a response saying that you have been added, you must pause here and wait until you receive it. You can repeat Step 12 tp check if you have been added to the Engineering group.
26. Open a terminal and enter the following commands to set up your git config and to create a code folder. The **code** folder is standard practice at Pinterest and there are scripts that default to that directory.

    ```
    $ git config --global user.email <your_username>@pinterest.com
    $ git config --global user.name <your_username>
    $ mkdir ~/code
    $ cd ~/code
    $ git clone --depth 1 ssh://git@phabricator.pinadmin.com/diffusion/P/pinboard.git
    ```

    If you are prompted to accept the XCode license agreement and enter your password, use the password that you created for your user account for your Macbook.

    The last command you ran is now cloning Pinboard to your local repository. This can take up to twenty minutes, depending on your Internet speed.

    If you get no errors after twenty minutes, you are ready for the first Basecamp session. If you receive any errors, follow the troubleshooting steps below.

› Troubleshooting locally cloning Pinboard

If you see **permission denied**:

- Check that you've completed steps 6 through 12 above and that you have been added to the Engineering project.

If you see **public key error**:

- Check that you've completed steps 13 through 25 above to set up SSH authentication.

This is the end of Basecamp Computer Setup prerequisites

You will walk through the rest of the steps on this page guided by the instructor during the first Basecamp session.

Computer Setup Lab

Welcome to the Basecamp Computer Setup Lab. These are the steps that you will now run through guided by the instructor of the first Basecamp Module: Computer Setup.

- As a prerequisite for these steps you must have completed the steps above titled **Pre-Lab Steps**. If you have not done this, tell your instructor.
- The following steps are grouped but must be followed in the order that they appear.

Phabricator Setup

Phabricator is what we use here for code review, see file history, repo commit history, as well as all the different repos we have here at Pinterest. To interface with Phabricator from our terminal, we need to set up Arcanist.

1. Arcanist should be installed by default on your Mac. You can check by running this command in a terminal on your local machine:

   ```
   $ which arc
   ```

   You should see this or something similar as the response:

   ```
   /Library/Application Support/Pinterest/CPE/pkgs/arcanist/bin/arc
   ```

   If you don't receive that response, request support in channel #phab in Slack.

2. Set up arc config to point to Pinterest:

   ```
   $ arc set-config phabricator.uri https://phabricator.pinadmin.com/
   ```

   You should see this response:

   ```
   Set key "phabricator.uri" = "https://phabricator.pinadmin.com/" in user config (was null)
   ```

3. Install arc certificate:

   ```
   $ arc install-certificate
   ```

   You should see this response:

   ```
   CONNECT Connecting to "https://phabricator.pinadmin.com/api/"...

   LOGIN TO PHABRICATOR

   Open this page in your browser and login to Phabricator if necessary:

   https://phabricator.pinadmin.com/conduit/login/

   Then paste the API Token on that page below.

       Paste API Token from that page:
   ```

4. Copy the API token from the Phabricator page and paste it into your terminal as prompted.

Prepare and SSH into Your Devapp

To create your ssh file and populate it, you can either use the terminal or create it manually using any Mac text editor.

The default terminal text editor for arcanist is nano. If you prefer a different editor, you can set it as shown in this example:

```
$ arc set-config editor vim
$ arc set-config editor "code -w -n"
```

The steps shown here use the Terminal and VIM:

1. Create your SSH config file by entering this command:

   ```
   $ touch ~/.ssh/config
   ```

2. Open the file for editing:

   ```
   $ vim ~/.ssh/config
   ```

3. Paste this template into the SSH config file:

   ```
   Host devapp
     HostName dev-NAME.ec2.pin220.com # Replace NAME with your username. Example: dev-
   xixia.ec2.pin220.com
     User NAME # Replace NAME with your username. Example: User xixia

   CheckHostIP no

   ServerAliveInterval 240
   ServerAliveCountMax 10
   ```

4. Replace **NAME** in the second and third lines with your ldap username, as shown in the example in the file.

5. Delete the comments in the file. (Comments shown in red above).

6. Save the file and exit the editor by pressing the **escape** key and then entering :wq

   You can now SSH into your devapp.

   NOTE: To SSH into your devapp you must have completed the pre-lab steps at the top of this section and received a response telling you that you've been added. The response can take around 30 min.

7. SSH into your devapp using gironde:

   ```
   $ gironde ssh devapp
   ```

   If your SSH succeeds, you see a response similar to this:

   ```
   Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1051-aws x86_64)

   ======= Pinterest AWS EC2 Node =======

   Last login: Fri Apr  3 20:56:59 2020 from 172.16.11.249
   xixia@dev-xixia ~ $
   ```

   NOTE: You must SSH into your devapp with gironde. If you try to use simply `$ ssh devapp` it will fail.

   NOTE: For SSH troubleshooting steps see this page.

Start Your Devserver

NOTE: All of the following commands are run on your devapp , after you've logged into it using SSH. They are not run on your local machine.

1. Make sure docker-compose is installed. Check the installation by using **which**:

   ```
   $ which docker-compose
   ```

The response should be:

```
/usr/local/bin/docker-compose
```

If you get a response that tells you that docker-compose cannot be found, install it as follows:

```
$ apt-get install docker-compose
```

If you get an error about root permissioning, add sudo:

```
$ sudo apt-get install docker-compose
```

2. cd into the code directory in your devapp:

   ```
   $ cd ~/code
   ```

   NOTE: If the **code** directory does not exist on your devapp, create it, then cd into it.

   ```
   $ mkdir ~/code
   $ cd ~/code
   ```

3. Clone the pinboard repo onto your devapp:

   ```
   $ git clone --depth 1 ssh://git@phabricator.pinadmin.com/diffusion/P/Pinboard.git pinboard
   ```

   NOTE: This clone will go much faster than it did in the pre-lab steps when you cloned to your local machine.

4. cd into the pinboard directory:

   ```
   $ cd ~/code/pinboard
   ```

5. Run **make web-api** from the pinboard repository root directory:

   ```
   $ make web-api
   ```

   Let this run to completion. It will take around fifteen minutes on the first run.

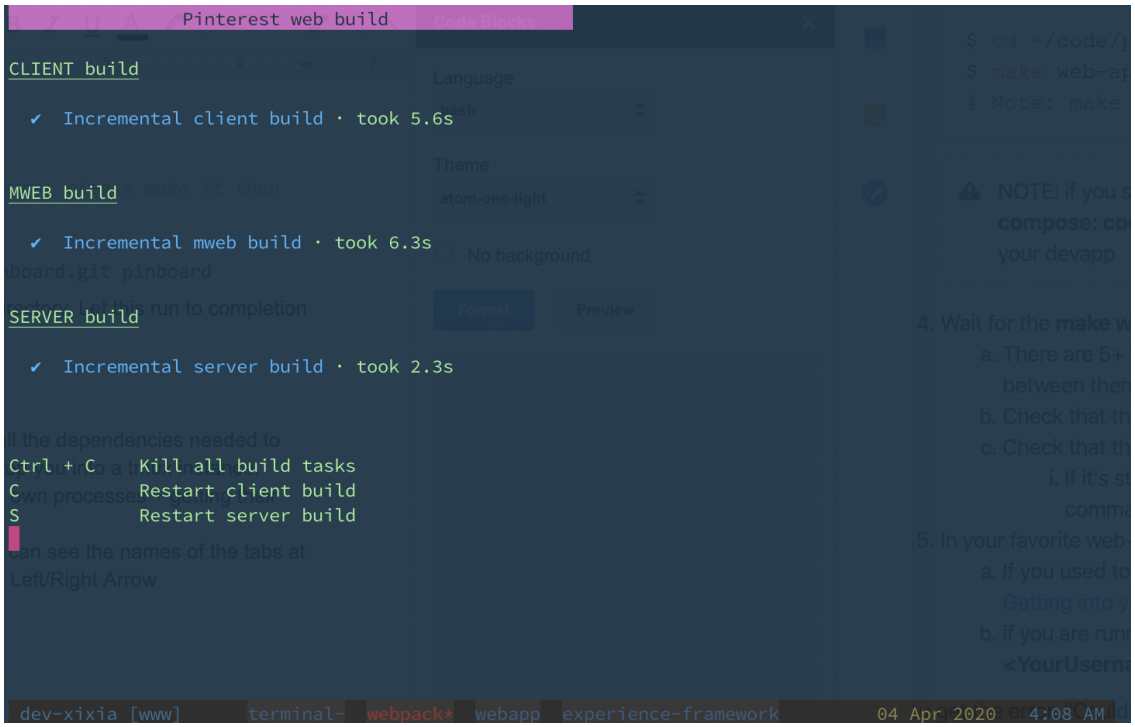
   The `make web-api` command sets up a docker container with all dependencies needed to run a local devserver for your webapp.

   When this is complete, you are dropped into a tmux instance, which has multiple tabs. Each of these tabs runs its own process to get its dependencies, wait and let the processes run their course.

   You can see the names of the tabs at the bottom.

6. Navigate to the **webpack** tab by pressing SHIFT + Left/Right Arrow

   When you see two or more check marks as shown below, this means that processes have completed.



7. In your Web browser go to:

   `<yourusername>.pinterdev.com`

   For example: xixia.pinterdev.com

   You should see the www.pinterest.com webapp.

   If it does not load successfully, navigate to the ngapi tab in your tmux instance for an error message, for example `command nodemon not found`.
   Try running the api process by hitting your up arrow to load the last command and pressing enter. Let that run until you see a message like `The API is ready.`

8. To prepare for your next Basecamp, follow the steps here to set up watcher

No labels