CSR to Github Migration

## **Introduction**

Cloud Build enables you to create triggers on a GitHub Enterprise instance. This page explains how you can use GitHub Enterprise triggers to invoke builds in response to commits or pull requests from a GitHub Enterprise instance and explains how you can build repositories from GitHub Enterprise if your instance is hosted in a private network.

High-level steps:

| Port opening for Azure |
| --- |
| Port opening for GCP(interconnect) |
| Create 15 repos(DocAI) on Github |
| Grant access to 40 members |
| Clone and migrate 15 repos from CSR to Github |
| [Update the cloud build triggers](https://cloud.google.com/build/docs/automating-builds/github/build-repos-from-github-enterprise-private-network) (44 triggers) |
| Test each cloud build trigger |

## **Before you begin**

* Enable the Cloud Build, Secret Manager, and Service Networking APIs.  
  [Enable the APIs](https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapis.com,secretmanager.googleapis.com,servicenetworking.googlepis.com&redirect=https://cloud.google.com/build/docs/automating-builds/github/build-repos-from-github-enterprise-private-network)
* Follow the instructions to [connect a GitHub Enterprise host](https://cloud.google.com/build/docs/automating-builds/github/connect-host-github-enterprise).
* Follow the instructions to [connect a GitHub Enterprise repository](https://cloud.google.com/build/docs/automating-builds/github/connect-repo-github-enterprise).

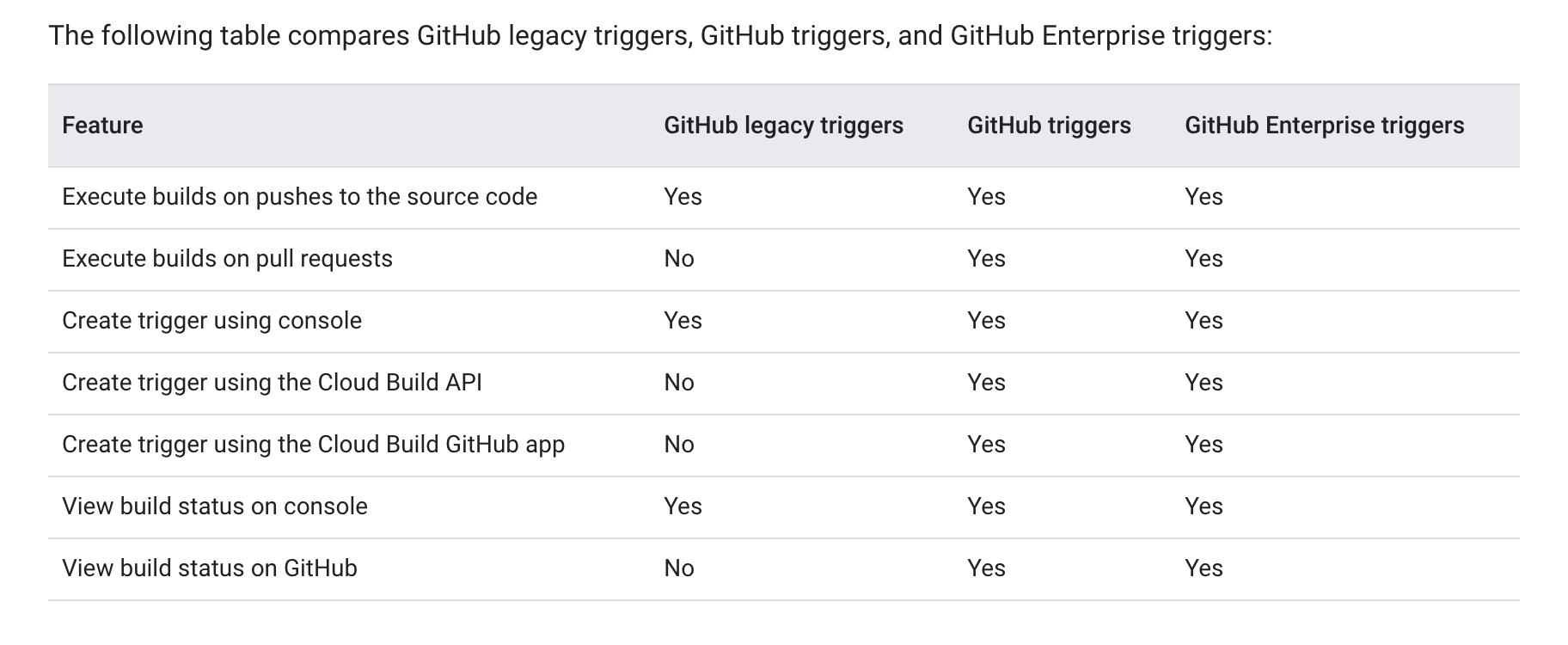
## **Building repositories from GitHub Enterprise in a private network**

If your GitHub Enterprise instance is hosted in a private network and not reachable over a public internet connection, you must create a private connection between your VPC network and the [service producer network](https://cloud.google.com/vpc/docs/private-services-access#service_producer_network). You also need a [private pool](https://cloud.google.com/build/docs/private-pools/private-pools-overview) to build in a private network.

To create a GitHub Enterprise trigger to build in a private network:

1. Create a private connection between the VPC network and your service producer network by completing the following steps:
   1. [Create a new VPC network](https://cloud.google.com/vpc/docs/create-modify-vpc-networks#create-custom-network) or [select an existing VPC network](https://console.cloud.google.com/networking/networks/list).  
      **Note:** You must configure your VPC network to have access to your GitHub Enterprise instance in a private network.
   2. [Allocate a named IP range in the VPC network](https://cloud.google.com/vpc/docs/configure-private-services-access#allocating-range). To use the VPC network with Cloud Build, your prefix length must be /23 or lower, such as /22, /21, etc.
   3. Establish a VPC network private connection between your VPC network and Google Cloud. To learn more, see [Creating a private connection](https://cloud.google.com/vpc/docs/configure-private-services-access#creating-connection).
   4. If you have DNS configured for your GitHub Enterprise instance, you need to manually peer your DNS zone to our service provider. To learn more, see the [Sharing private DNS zones with service producers](https://cloud.google.com/vpc/docs/configure-private-services-access#dns-peering).
   5. [OPTIONAL] If you do not want to create a private connection from your Cloud project, you can set up a [Shared VPC](https://cloud.google.com/vpc/docs/provisioning-shared-vpc) and have your Cloud project use that network instead if your project is part of an organization. You will still need the Shared VPC to be peered to the network.
2. Use [private pools](https://cloud.google.com/build/docs/private-pools/private-pools-overview) to run your builds. If you have not created a private pool, see [create a new private pool](https://cloud.google.com/build/docs/private-pools/create-manage-private-pools).
3. [Create a GitHub Enterprise trigger](https://cloud.google.com/build/docs/automating-builds/build-repos-from-github-enterprise#creating_a_github_trigger_for_your_github_enterprise_installation) to build repositories hosted on a GitHub Enterprise instance.

Your GitHub Enterprise trigger will now automatically invoke builds on your GitHub Enterprise instance based on your configuration. To learn how to run builds using a private pool, see [Running builds in a private pool](https://cloud.google.com/build/docs/private-pools/run-builds-in-private-pool).

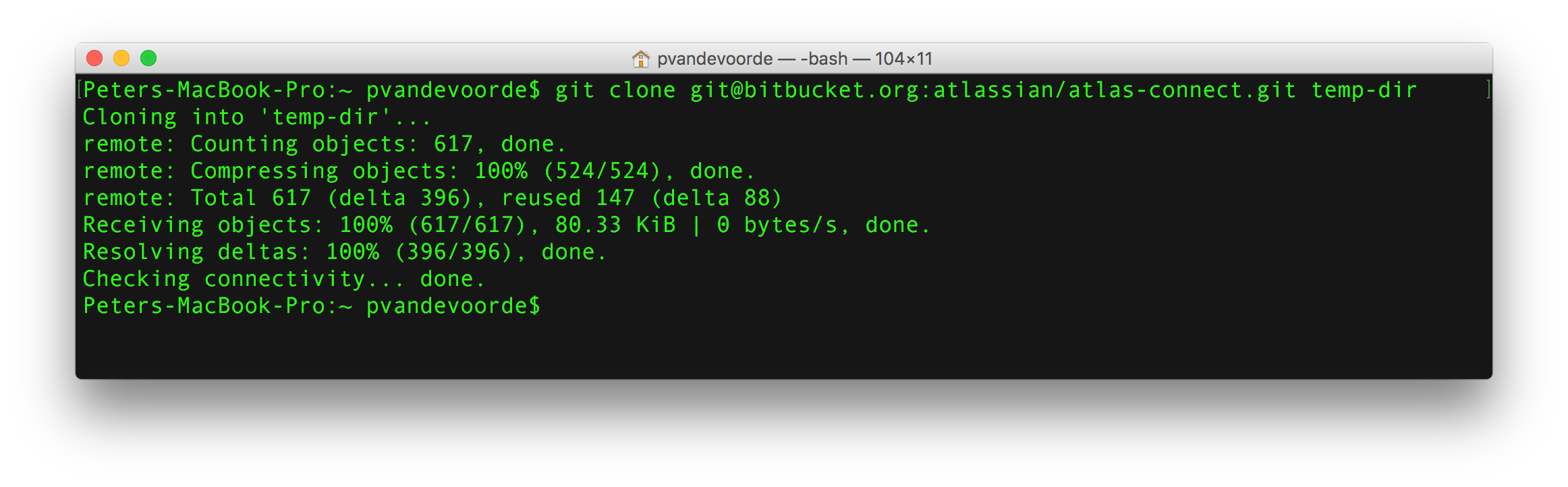


## **II. Migrate code from CSR to Github**

Let’s call the original repository ORI and the new one NEW, here are the steps required to copy everything from ORI to NEW:

1. Create a local repository in the **temp-dir** directory using:

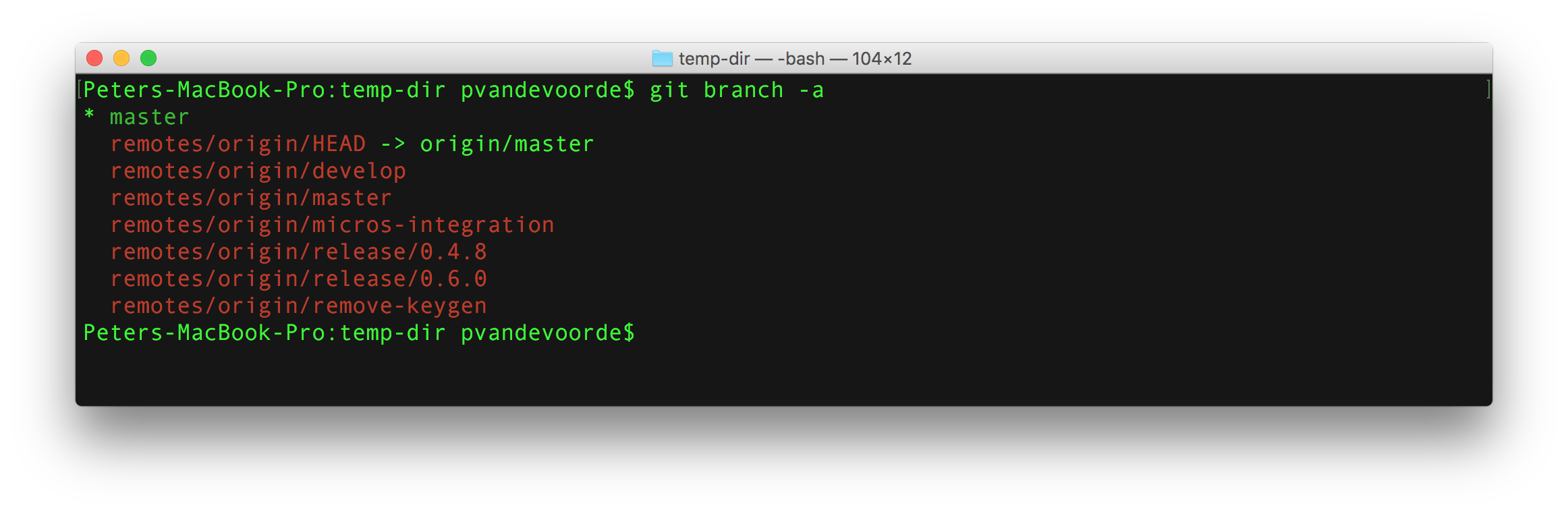
git clone <url to ORI repo> temp-dir



2. Go into the temp-dir directory.

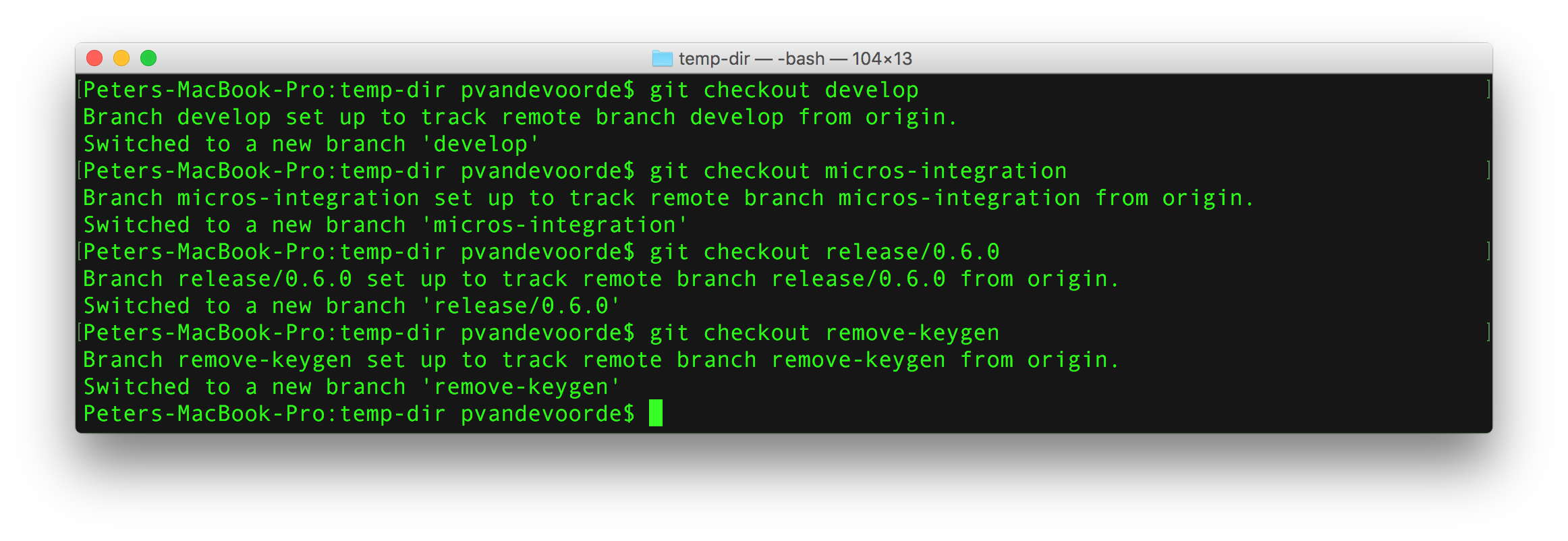
3. To see a list of the different branches in ORI do:

git branch -a



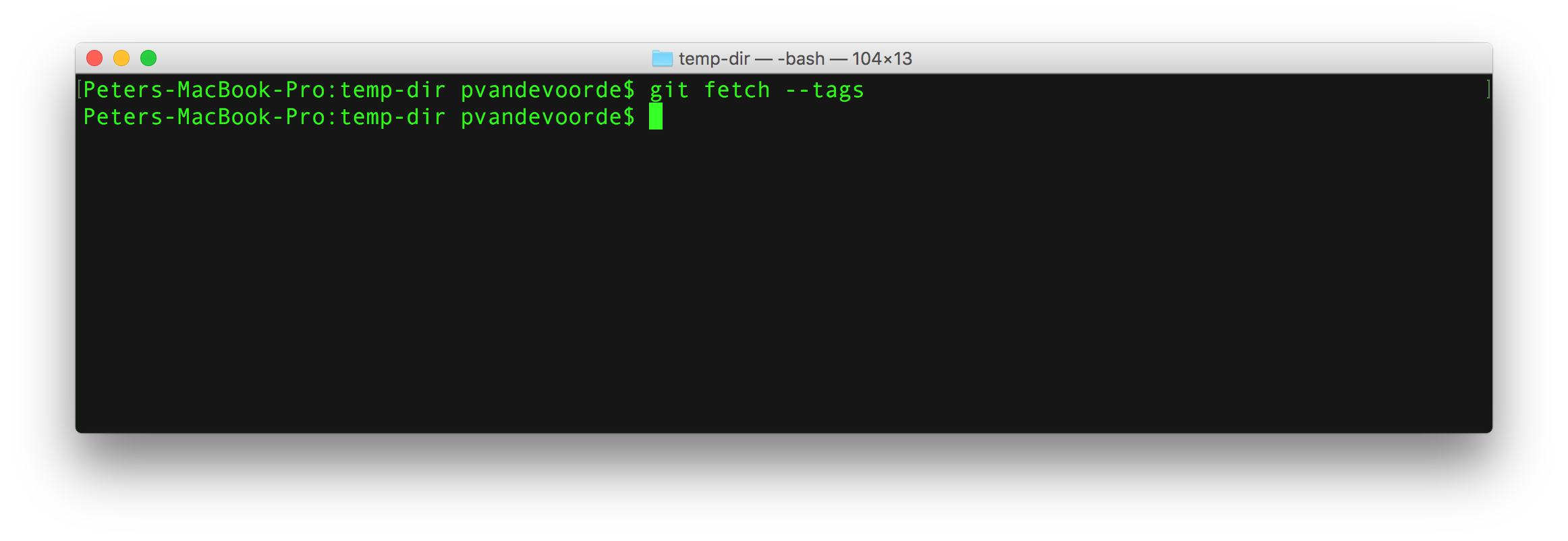
4. Checkout all the branches that you want to copy from ORI to NEW using:

git checkout branch-name



5. Now fetch all the tags from ORI using:

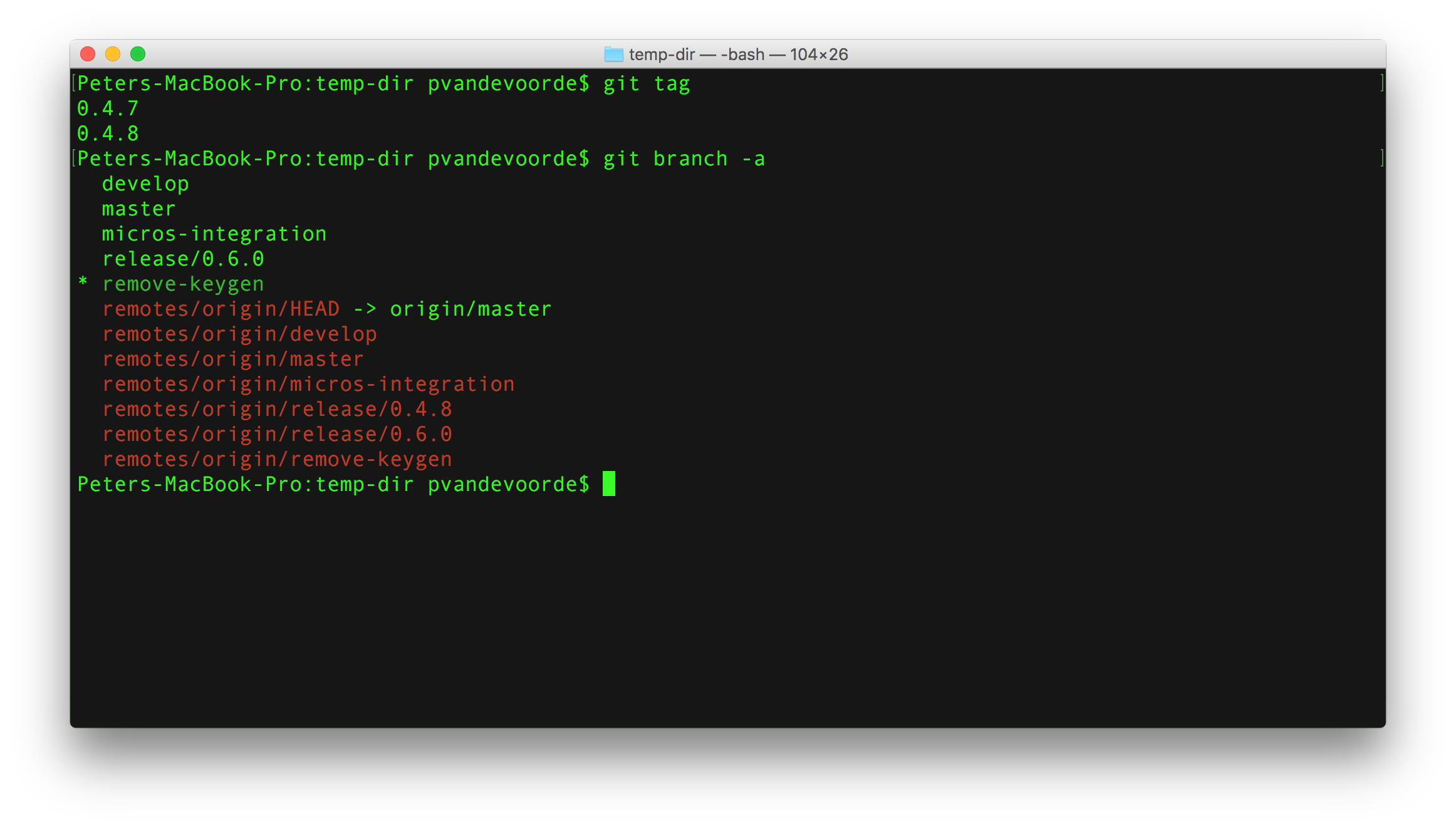
git fetch --tags



6. Before doing the next step make sure to check your local tags and branches using the following commands:

git tag

git branch -a



7. Now clear the link to the ORI repository with the following command:

git remote rm origin

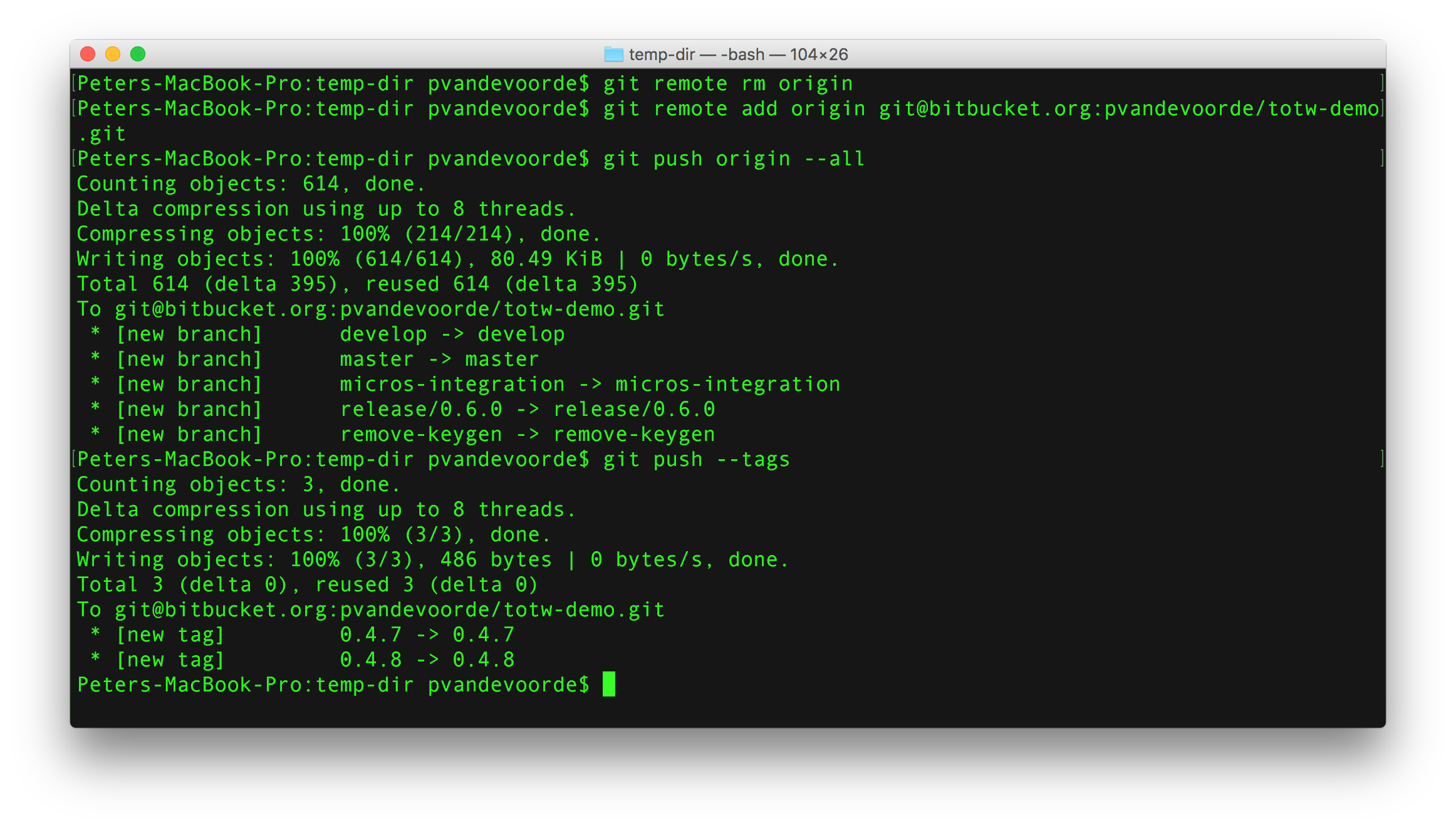
8. Now link your local repository to your newly created NEW repository using the following command:

git remote add origin <url to NEW repo>

9. Now push all your branches and tags with these commands:

git push origin --all

git push --tags



10. You now have a full copy from your ORI repo.

## 

## 

## **III. Update Cloud Build Triggers to use Github URLs**

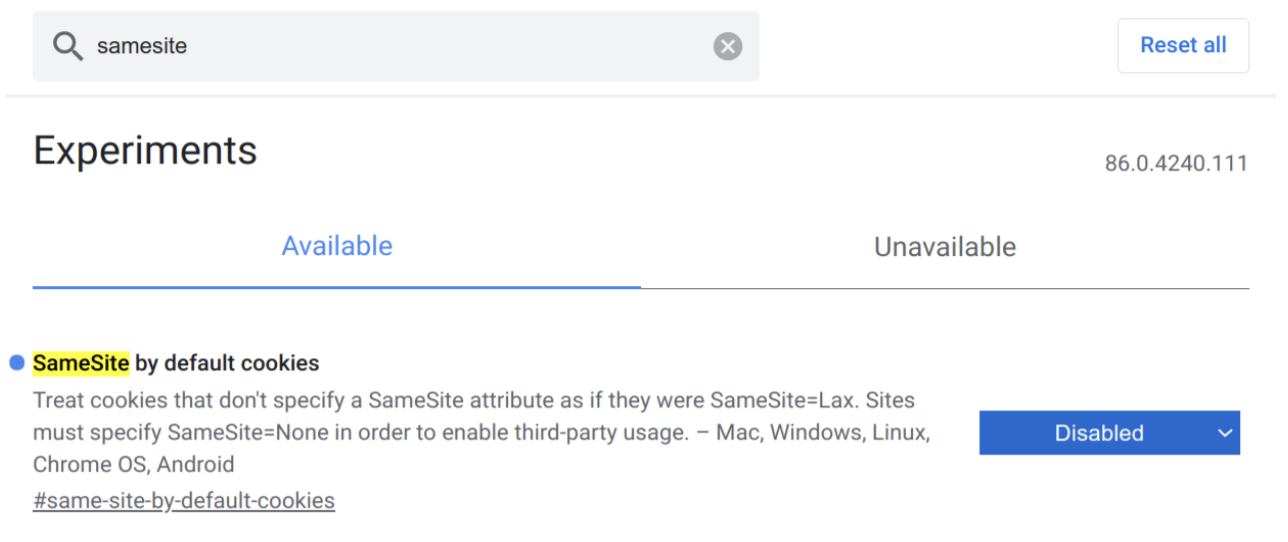
## **Before you begin**

* Enable the Cloud Build and Secret Manager APIs.  
  [Enable the APIs](https://console.cloud.google.com/flows/enableapi?apiid=cloudbuild.googleapis.com,secretmanager.googleapis.com&redirect=https://cloud.google.com/build/docs/automating-builds/github/build-repos-from-github-enterprise)
* Follow the instructions to [connect a GitHub Enterprise host](https://cloud.google.com/build/docs/automating-builds/github/connect-host-github-enterprise).
* Follow the instructions to [connect a GitHub Enterprise repository](https://cloud.google.com/build/docs/automating-builds/github/connect-repo-github-enterprise).

## **Connecting to a GitHub Enterprise host**

You need to create a GitHub application on your GitHub Enterprise instance. The app sends webhook events to a Cloud Build endpoint. Upon receiving these events, Cloud Build will validate the payload and execute a build if the event corresponds to a Cloud Build GitHub trigger.

This sections explains how you can create a GitHub app:

1. Log in to your GitHub Enterprise instance.
2. Ensure you have [latest version](https://docs.github.com/en/enterprise-server@2.22/admin/enterprise-management/upgrading-github-enterprise-server) of GitHub Enterprise installed.  
   Some versions of GitHub Enterprise may require SameSite cookies to be disabled in order to complete the following steps in a Chrome browser. If you are a version of GitHub Enterprise prior to the 2.21.3 release, you will need to disable SameSite cookies:
   1. Go to chrome://flags/.
   2. Type samesite in the filter bar.
   3. Make sure SameSite by default cookies is DISABLED.  
      
   4. Restart your browser.
3. **Note:** The latest version of GitHub Enterprise does not require you to complete these steps.
4. Open the Cloud Build **Manage repositories** page:  
   [Open the Manage Repositories page](https://console.cloud.google.com/cloud-build/repos/)
5. Click **Connect host**.  
   You will see the **Connect host** panel, which prompts you to create a host connection to connect your GitHub Enterprise repositories to Cloud Build.
6. In the **Host URL** section, enter the URL for your GitHub Enterprise instance. For example, ghe.example.com.
7. In the **API key** section, click **Generate** to generate an API key or enter an API key if you already have one.  
   If you want to manually create an API key, complete the following step:  
   To obtain an API key:
   1. Open the **Credentials** page in the Cloud console:  
      [Open the Credentials page](https://console.cloud.google.com/apis/credentials)
   2. Click **Create credentials**.
   3. Click **API Key**.  
      You will see a pop-up box with your API key created.
   4. Click **Restrict key**.
   5. Under **API Restrictions**, select **Cloud Build API** from the drop-down menu.
   6. Click **Save**.
8. **Note:** You can re-use the same API key for multiple connections.
9. [OPTIONAL] In the **Organization** section, enter the organization the GitHub app will be created for. If this section is left blank, the app will be created under the current user account.
10. [OPTIONAL] In the **CA Certificate** section, click **Browse** to upload your self-signed certificate. Your certificate must not exceed 10 KB in size and should be in PEM format (.pem, .cer, or .crt). If this section is left blank, a [default set of certificates](https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/#5-certificates) will be used in place.  
    **Note:** By default, Google only trusts a set of publicly trusted certificate authorities. If you want to change the certificate authorities to trust or if your certificate is self-signed, you can provide your own set of certificates to trust.
11. [OPTIONAL] In the **Network** section, enter the name of your **Network project** and a **Network name** for your network if your GitHub Enterprise instance is hosted on-prem and you have [peered your network](https://cloud.google.com/build/docs/automating-builds/create-github-enterprise-triggers#peering_a_vpc_network) to Google Cloud.
12. Click **Connect Host**.  
    If your GitHub Enterprise instance is in a private network, the host connection process may take several minutes to complete.
13. If you want to connect your repositories to Cloud Build, click **Connect Repositories**. Otherwise, click **Done**.
14. After you connect your host, a pop-up box will appear prompting you to enter the name of your GitHub Enterprise app. Prior to entering the name of your app, you may be asked to log in. If you are using Google Chrome as your browser, the pop-up page may ask you to enter information about your GitHub Enterprise app manually.
15. After logging in, enter a name for your GitHub app.
16. Click **Create GitHub App**.  
    You have just created a GitHub app on your GitHub Enterprise instance. Cloud Build will automatically store your credentials in [Secret Manager](https://console.cloud.google.com/security/secret-manager) and connect the host to your Cloud Project. In the API, this connection is represented as a GitHubEnterpriseConfig resource, or an association between Cloud Build and your GitHub Enterprise Server.  
    **Note:** This step attempts to store secrets in Secret Manager with a [replication policy](https://cloud.google.com/secret-manager/docs/locations) with a global scope. If there are any organization-level resource location constraints, this step will not succeed.  
    Your host is now successfully connected. You can click on **Connect Repositories** if you want to connect repositories to Cloud Build. To learn more, see [Connect to a GitHub Enterprise repository](https://cloud.google.com/build/docs/automating-builds/github/connect-repo-github-enterprise).

**Note:** When creating a GitHub app following these steps, the app is created for your GitHub user account. If you want to make the app visible to other users, you will need to transfer ownership of the app to a different user account or organization.

## **Connecting to a GitHub Enterprise repository**

This section explains how you can map your Cloud project to the GitHub app you created. If you have not created the app yet, follow the instructions in [Connect to a GitHub Enterprise host](https://cloud.google.com/build/docs/automating-builds/github/connect-host-github-enterprise).

[Console](https://cloud.google.com/build/docs/automating-builds/github/connect-repo-github-enterprise#console)

[API](https://cloud.google.com/build/docs/automating-builds/github/connect-repo-github-enterprise#api)

1. Open the **Triggers** page in the Google Cloud console.  
   [Open the Triggers page](https://console.cloud.google.com/cloud-build/triggers)
2. Click **Connect Repository**.
3. Under **Select source**, click on **GitHub Enterprise**.
4. Select your **Host Connection** (GitHub Enterprise Config) from the drop-down menu.
5. Click **Continue**.
6. Authorize your application.
7. Under **Select repository**, select your **GitHub account** and **Repository** from the drop-down menu.