

## Denoto Deployment Instructions

Denoto is a two-part system, consisting of a server back-end and an in-browser client. Since Denoto makes use of cutting-edge web technologies, Google Chrome is the only browser we are able to support at this time.

The server has been implemented using Ruby on Rails and PostgreSQL. Throughout our development and testing, we have used Ubuntu 14.04 as our server platform. While we would expect Denoto to be deployable on any server platform that supports Ruby and PostgreSQL, we strongly advise Ubuntu and cannot provide specific deployment instructions for other platforms.

### 1) Installing Ruby & Rails

Denoto requires Ruby with a minimum version of 2.0 and Rails with a minimum version of 4.0.

To install the latest stable version of Ruby on Rails, we recommend using the command-line tool RVM (Ruby Version Manager) in Ubuntu. The following command is the easiest way to install the latest version:

```
$ \curl -sSL https://get.rvm.io | bash -s stable --rails
```

### 2) Setting up the PostgreSQL Database

#### 2.1) Installing

If you don't have PostgreSQL installed already, you can use your platform's package manager to install it. This is the command in Ubuntu:

```
$ sudo apt-get install postgresql libpq-dev phpPgAdmin pgAdmin3
```

#### 2.2) Log in to the postgres Prompt

Before creating or modifying your PostgreSQL database, you must first create a superuser account. The package installation creates a default user account called postgres which you can use to log in to the postgres service. Log in to that account by typing the following command:

```
$ sudo su postgres -c psql
```

#### 2.3) Create a New User

After you have logged in to the postgres service, create a new user by entering the following command:

```
# create user <username> with password '<password>';
```

Replace <username> and <password> with your own account details. Note that the semicolon is required to complete the command.

#### 2.4) Make the New User a Superuser

In order to get permissions to create, modify, and remove databases, you must first upgrade your new user to superuser status:

```
# alter user username superuser;
```

#### 2.5) Exit the postgres terminal & Create User Database

Now that you have created a new superuser account, enter '\q' to exit the postgres interactive terminal. When you are back in the Linux shell, run the command 'createdb' to create a default database for your new user.

## 2.6) Install Ruby Gems

Denoto makes use of a number of libraries, also known as Ruby Gems. You can view the dependencies by viewing the contents of the file Gemfile found in the root directory of the Denoto repository. You can install the dependencies by running the following shell command from the root directory of the Denoto repository:

```
$ bundle install
```

## 2.7) Create the Application Database

The Denoto repository contains schema files for development, testing, and production databases. The following shell command will create new databases in the directories db/denoto\_development, db/denoto\_test, and db/denoto\_production:

```
$ rake db:create:all
```

## 2.8) Grant Database Permissions to the User

In order to modify the new databases, you must grant your user account privileges. First, log in to the postgres service by running the 'psql' command in the shell. Once you are logged in to the postgres terminal, run the following commands:

```
# grant all privileges on database "db/denoto_development" to username;
# grant all privileges on database "db/denoto_production" to username;
# grant all privileges on database "db/denoto_test" to username;
```

As before, you can exit the postgres terminal by entering '\q' into the prompt.

## 3) Deploying Denoto on the web

### 3.1) Install the Application Server

This command will download and install passenger, Denoto's application server.

```
$ gem install passenger
```

### 3.2) Install web server

Passenger includes with a utility to install Nginx, the web server used in Denoto. Run the following shell command to start the interactive installer:

```
$ passenger-install-nginx-module
```

In the installer, select Ruby as your choice of language and choose option 1, "Yes: download, compile, and install Nginx for me."

### 3.3) Configuring Nginx

Open the provided nginx.conf file found in /opt/nginx/conf/nginx.conf. Inside the server block on line 38, change root to be the root directory of your Denoto repository. Reload the updated configuration with the following shell command:

```
$ sudo service nginx reload
```

### 3.4) Start the Web Server

Now that the server has been configured, restart it by running the following shell command:

```
$ sudo service nginx start
```

The server should be available on port 80.

### 3.5) Start a Local Server

For testing purposes, it can be easier to use a local development server. To start the local server, navigate to the root directory of the Denoto repository and run this command:

```
$ rails s
```

The development server should be available on port 3000 of your localhost.