
Flowcell Tool Documentation

Release 0.1.0

Manuel Holtgrewe

Mar 23, 2018

Installation Getting Started

1	Dependencies	3
2	Features	5

Flowcelltool is a Django web application for the management of Illumina flow cells. The documentation is split into three parts (accessible through the navigation on the left):

Installation & Getting Started Instructions for the installation of the web application and its deployment

Manual This section contains the user documentation

Project Info More information on the project, including the changelog, list of contributing authors, and contribution instructions.

CHAPTER 1

Dependencies

- Python 3
- Django 1.10
- PostgreSQL

- Graphical management of flow cells and libraries
- Automated generation of bcl2fastq (both v1.x and v2.x) sample sheets
- Authentication via LDAP/ActiveDirectory or local users
- Easily deployable to Heroku/Flynn.io/Docker (12 factor app), follows Two Scoops of Python best practice

The Project repository can be found on [Github.com](#).

2.1 Installation Overview

Flowcelltool has been developed using the [The Twelve-Factor App](#) principles. This means that deployment is easily possible on a wide range of platforms, including “classic” virtual machines as well as PAAS servers.

Here, the following installation options are documented:

Install on Heroku Heroku is a platform as a service (PAAS) provider. Flowcelltool can be installed as a “free plan” application with a few clicks and without software to install on your local computer.

Install on Flynn Flynn is an open source effort for easily providing self-hosted PAAS services, similar to Heroku.

Install using Ansible Install on a virtual machine. We document the necessary steps in an Ansible playbook. You can look at the Ansible playbook to figure out the steps for a manual installation.

Install Locally (Dev) Create a local setup for the development of Flowcelltool.

After installation, refer to [Getting Started](#) for information on getting the big picture.

2.2 Install on Heroku

Heroku is a platform as a service (PAAS) provider. Flowcelltool can be installed into their free plan option with a few clicks. This is the easiest installation option and for trying out Flowceltool with minimal effort.


2.2.1 Prerequisites

Create a (free) account on [Heroku](#) and log into the account.

2.2.2 Installation


Click on the following button to start the installation. You will be directed to the Heroku website.

Fill in the name for your Flowcelltool installation and click on the “Deploy App” button.




Flowcelltool
Simple management of Illumina sequencing flowcells
[bihealth/flowcelltool#features/deploy-to-heroku](#)


1 App name

foo-flowcelltool 

foo-flowcelltool is available


Choose a region

Europe 

 Add to pipeline...

Add-ons

These add-ons will be provisioned when the app is deployed.

 Heroku Postgres Hobby Dev (Free)

Config Variables

DJANGO_SETTINGS_MODULE Required

The Django environment to use

config.settings.production

DJANGO_SECRET_KEY

A secret key for verifying the integrity of signed cookies.

autogenerated

DJANGO_ALLOWED_HOSTS Required

Allowed hosts

*

FLOWCELLTOOL_INITIAL_ROOT_PASSWORD

Initial root password

autogenerated

2 Deploy app

Wait until the installation is complete.

An administration user `root` will be automatically created with a random password. You can lookup the automatically generated password now.

- Click “Manage App” button displayed after successful installation in a new tab.
- Go to “Settings”
- Click “Reveal Config Vars”
- The password for the `root` user is stored in the `FLOWCELLTOOL_INITIAL_ROOT_PASSWORD` config variable.
- Copy that password into the clipboard.
- Click on the “Open app” button on the top.
- Log in with user `root` and the password from the clipboard.

2.2.3 Continuing From Here

Now, continue with the *Getting Started* guide or read on Email sending and LDAP authentication in *Configure Advanced Features*.

2.3 Install on Flynn

Flynn is a PaaS system similar to Heroku that you can run on your own hardware.

2.3.1 Prerequisites

Start by installing Flynn on your server and installing the `flynn` command line on your local machine as described in the *Flynn manual: Installation*.

2.3.2 The Actual Deploying

First, clone the repository from Github and get the latest stable version.

```
$ git clone git@github.com:bihealth/flowcelltool.git
$ git checkout v0.1.0
```

Then, create a new Flynn app

```
$ cd flowcelltool
$ flynn create flowcelltool
Created flowcelltool
```

Next, provision a PostgreSQL database

```
$ flynn resource add postgres
Created resource d5d9350d-b55e-4102-a9d3-b5d4bbbd987c and release 56857385-d3ae-4c7e-
↪8259-7fb2e184e064.
```

Create a Redis database for caching

```
$ flynn resource add redis
Created resource ba6187e7-1fed-4cbl-ae3f-d9f719d1ce69 and release 83e8b2da-9cc0-4c25-
→8668-a07c09493a55.
```

Ensure that the Flowcelltool Django app uses production settings.

```
$ flynn env set DJANGO_SETTINGS_MODULE=config.settings.production
```

Set the Django key to something secret and set DJANGO_ALLOWED_HOSTS.

```
$ pwgen 100 1
# ensure some random string is printed
zaeFahB5oot3aiciegoheil0iSeis0ufahChaeveuJumi3sai8sheequ6weewetushe7jei6veiBohhaiphoe felu0Eiy1nae3S
$ flynn env set DJANGO_SECRET_KEY=$(pwgen 100 1)
$ flynn env set DJANGO_ALLOWED_HOSTS='*'
```

Finally, deploy the application

```
$ git push -u flynn master
```

Setup database using migrate

```
$ flynn run /app/manage.py migrate
```

Create a superuser

```
$ flynn run /app/manage.py createsuperuser
```

Then follow the instructions of the `createsuperuser` command.

2.3.3 Continuing From Here

Now, continue with the *Getting Started* guide or read on Email sending and LDAP authentication in *Configure Advanced Features*.

2.4 Install using Ansible

Ansible is a software for the management of servers and can be used for automatically deploying Flowceltool.

You can also easily infer the steps for manual deployment from the Ansible playbook.

2.4.1 Prerequisites

1. You have setup a virtual machine with CentOS 7.4 Linux. (Of course, any modern Linux will work but will need adjustments to the Ansible playbook file).
2. You can connect to the VM as user root with SSH (i.e., you properly setup the `authorized_keys` for root and configured the SSH server appropriately).
3. You have Ansible (>=2.4) installed on your local machine (e.g., using `pip install ansible`).
4. You have the `pwgen` binary installed on your local machine.

2.4.2 Installation

First, clone the repository from Github and get the latest stable version.

```
$ git clone git@github.com:bihealth/flowcelltool.git
$ git checkout v0.1.1
```

Create an inventory file in the `ansible` sub directory with the remote server's hostname.

Note that we use Ansible variables here to set the name and password of the Postgres user that Flowcelltool will use. Also note that you can change the Flowcelltool version here and the automatically created super user name (default: `root`) and password (default: `"password"`). In a more refined Ansible setup, you would use vault-encrypted host variables.

```
$ cat <<EOF >inventories.yml
---
flowcelltool-servers:
  hosts:
    "your-vm-hostname":
      # Postgres Configuration
      #
      # Postgres database to create
      FLOWCELLTOOL_DB: 'flowcelltool'
      # User to use for connecting to postgres server
      FLOWCELLTOOL_PG_USER: 'flowcelltool'
      # Password of user defined above
      FLOWCELLTOOL_PG_PASSWORD: 'flowcellpass'

      # Flowcelltool Configuration
      #
      # Version of Flowcelltool to install
      FLOWCELLTOOL_VERSION: 'stable'
      # Super user name to create
      FLOWCELLTOOL_SUPERUSER: root
      # Super user password to set
      FLOWCELLTOOL_SUPERUSER_PW: password
      # Generate secret key
      DJANGO_SECRET_KEY: '`pwgen -N 1 40`'
EOF
```

Now, execute the Ansible playbook.

```
$ ansible-playbook -i inventory.yml inst-flowcelltool-centos.yml
```

Ansible playbooks are easy to read! If you want to find out how to install Flowcelltool manually.

Note that this will setup a PostgreSQL database, Nginx as a reverse proxy, and the Flowcell app itself. However, also note that it will only perform a setup for HTTP on port 80 and not yet an HTTPS server.

You can then go to `https://your-vm-hostname` and login with the user and password configured above. Of course, you will have to confirm the security exception for the self-signed SSL certificate.

2.4.3 Continuing From Here

Now, continue with the *Getting Started* guide or read on Email sending and LDAP authentication in *Configure Advanced Features*.

2.5 Install Locally (Dev)

This section describes how to setup Flowcelltool in a virtualenv in a development environment.

2.5.1 Prerequisites

1. Install and configure Postgres.
2. Create a user and database for the application in the database. For example, `flowcelltool_user` with password `flowcelltool_user` and a database called `flowcelltool`. Also, give the user the permission to create further Postgres databases (used for testing).

You have to make the credentials in the environment variable `DATABASE_URL`:

```
$ export DATABASE_URL='postgres://flowcelltool_user:flowcelltool_user@127.0.0.1/
↪flowcelltool'
```

3. Install Python 3 (>= 3.4)

2.5.2 Installation

Next, clone the repository and setup the virtual environment inside

```
$ git clone https://github.com/bihealth/flowcelltool.git
$ virtualenv -p python3 .venv
$ source .venv/bin/activate
```

Then, install the dependencies

```
$ pip install --upgrade pip
$ for f in requirements_*.txt; do pip install -r $f; done
```

Now, you can run the tests

```
$ python manage.py test
```

Then, initialize the database:

```
$ python manage.py migrate
```

Finally, start the server

```
$ python manage.py runserver
```

2.5.3 Continuing From Here

Now, continue with the *Getting Started* guide or read on Email sending and LDAP authentication in *Configure Advanced Features*.

2.6 Configure Advanced Features

This section describes how to configure some advanced features:

- LDAP authentication
- Sending of emails

2.6.1 Outgoing Email Configuration

You have to set the SMTP server for outgoing mail using the environment variable `EMAIL_URL`

On Heroku

Simply configure `EMAIL_URL` “Config Variable” in the Heroku application configuration.

```
EMAIL_URL=smtp://post-office.example.com
```

On Flynn

```
$ flynn env set EMAIL_URL=smtp://post-office.example.com
```

On Manual / Ansible Deployment

You have to set the variable similar to `DATABASE_URL` in `/etc/systemd/system/flowcelltool.service`. When using Ansible, you best configure this in `templates/flowcelltool.service.j2`.

```
Environment="EMAIL_URL=smtp://post-office.example.com"
```

2.6.2 LDAP Configuration

Flowcelltool can use up to two LDAP servers (ActiveDirectory is also supported) for authentication users. The configuration of the second one is optional. For one server, you can either configure the server to user `username` for login or `username@DOMAIN` with a configurable domain.

To enable this for the first server, define the following environment variables (see *Outgoing Email Configuration* on the appropriate places for the different deployment targets).

The configuration of `AUTH_LDAP_USERNAME_DOMAIN` is optional when only using one server.

```
ENABLE_LDAP=1
AUTH_LDAP_BIND_DN='CN=user,DC=example,DC=com'
AUTH_LDAP_BIND_PASSWORD='password'
AUTH_LDAP_SERVER_URI='ldap://activedirectory.example.com'
AUTH_LDAP_USER_SEARCH_BASE='DC=example,DC=com'
AUTH_LDAP_USERNAME_DOMAIN='YOURDOMAIN'
```

For configuring the secondary LDAP server, use the following environment variables. The configuration of `AUTH_LDAP_USERNAME_DOMAIN` is **required** when using two servers.


```
export ENABLE_LDAP_SECONDARY=1
export AUTH_LDAP2_BIND_DN='CN=user,DC=example,DC=com'
export AUTH_LDAP2_BIND_PASSWORD='password'
export AUTH_LDAP2_SERVER_URI='ldap://activedirectory.example.com'
export AUTH_LDAP2_USER_SEARCH_BASE='DC=example,DC=com'
export AUTH_LDAP2_USERNAME_DOMAIN='YOURDOMAIN2'
```

Note that for users logging in via LDAP, the username must be in form of `username@YOURDOMAIN` if the `AUTH_LDAP*_USERNAME_DOMAIN` variable is set.

Note: If you alter the username domain configuration once the tool is in use, you must manually alter the user names already found in the Django Postgres database.

2.6.3 Login Message

You can specify a message to display on the login screen by setting the environment variable `LOGIN_MESSAGE`.

2.7 Getting Started

This section gives an overview of how the application works. Flowcelltool allows you to manage flow cells on libraries with a graphical and then create sample sheets for the Illumina demultiplexing software. To facilitate this, you will need to register your sequencing devices (for configuring dual indexing workflow used) and the adapter barcode sequence used.

The overall workflow after the initial installation is:

1. Create or import adapter barcode libraries
2. Register your sequencers
3. Manage the users in the database

Once this is complete, you can start

- Adding your flow cells.

2.8 Manage Barcodes

By default, the database of the web application is completely empty. The first thing to add is a few barcodes.

The interface for managing barcode sets can be reached through the “Barcodes” link at the top and it is self-documenting.

Commonly used barcodes are available for download in the [Flowcelltool Github repository](#).

Note: Note that the barcode sequences should be given in “**forward**” orientation. Flowcelltool will automatically reverse-complement the second index in the case of dual indexing if the sequencing machine uses the [Illumina Dual-Index Paired-End Sequencing Workflow B](#).

If you create your own barcode set then please share them by [creating a ticket in the Flowcelltool Github issue tracker](#).

2.9 Manage Sequencers

Also, there are no sequencing machines added by default.

The interface for managing sequencers can be reached through the “Sequencers” link at the top and it is self-documenting.

Some notes on the sequencer fields:

Vendor ID The ID of the device, e.g., ST-K00100 or NB501000.

Label A short name of the device, e.g., “HiSeq 4000 in Lab 101”

Machine Model The machine model.

Slot count Number of slots in the device (e.g., 1 for NextSeq 500 and 2 for HiSeq 4000).

Dual index workflow The dual indexing workflow as described in the [Illumina Indexed Sequencing Overview Guide](#)

The `Vendor ID` is also encoded in the flowcell run output directory name. This information will be used for automatically reverse-complementing the second index on dual indexing based on the device used for sequencing a flowcell.

2.10 Manage Flowcells

The interface for managing flowcells can be reached through the “Flow Cells” link at the top.

The first step is first registering the flow cell with the necessary meta information. The second step is then adding the library information.

Finally, you can export the flow cell sample sheet as:

- Illumina `bcf2fastq 1.x` sample sheet for older runs with RTA v1.x
- Illumina `bcf2fastq 2.x` sample sheet for older runs with RTA v2.x
- YAML-based sample sheets for use in `cubi_demux`.

2.10.1 Creating Flow Cells

You can create a new flow cell using the “Creat New” button on the Flow Cell management site.

Some details on the flow cell meta data fields:

Name The name of the flowcell, e.g., 160303_ST-K12345_0815_A_BCD EFGHIXX_LABEL. This follows the format convention `${Date:YYMMDD}_${Machine_ID}_${Run_No}_${Slot}_${Flowcell_ID}[_${Label}]`.

Num Lanes The number of lanes on the flow cell, e.g., 8 for HiSeq, 4 for NextSeq.

Status The flow cell status:

initial meta data record, sequencing not started

sequencing complete sequencing is complete

sequencing failed sequencing failed

demultiplexing complete demultiplexing is complete

demultiplexing started demultiplexing has started

demultiplexing results delivered demultiplexing results (FASTQ) have been delivered as requested

base calls delivered raw base calls have been delivered as requested

Sequencer Operator The operator on the sequencer (free text)

Demultiplexing Operator The operator for demultiplexing, must be a user in the Flowcelltool database

Index read Count Number of index adapters, 0 for no multiplexing, 1 for one adapter, 2 for two adapters.

RTA version The RTA version used (matching `bcl2fastq` version will be used)

Read length The length of the sequencing reads as configured.

2.10.2 Attaching Messages and Files

Once you have created a flowcell, you can add messages and attach files. For example, this can be done for keeping a record of the sample sheets XLS files as received from the sequencing facility or attaching the QC report.

2.10.3 Copy-and-paste Excel Data

After creating a flow cell, you can easily add data using copy and paste from Excel, using `Actions -> Copy & Paste XLS`. This will start a wizard that works as follows.

1. On the first wizard screen, copy and paste the sample information data from your sample sheet. This should contain at one column for each
 - the name of the sample
 - the name of the first barcode (as defined in the barcode set)
 - the column describing the lane (e.g., as 1-4 or 1, 2, 4, 5)

Optionally, you can also add the name of the second barcode as well for dual indexing.

2. On the second screen, you can select the column for the sample, the column of the first barcode name, the first barcode set used (optionally also for the second barcode). Also, you should specify the number of the first data row and the column with the lanes that the library appears on. At the bottom, the pasted TSV data is previewed with column and row number.
3. Finally, the resulting data is previewed back to you for a last check and you can then store the libraries for the flow cell.

If you want to use the wizard with different barcode sets, you have to follow the process with each barcode set used.

Note that the barcode will be selected by the given name in the Excel sheet. This will be done using a “longest matching suffix” rule, such that, e.g., for Agilent Agilent SureSelect V6 barcodes, the number “12” will be matched to “A12”.

2.11 User Management

This is provisional information and needs some work.

User management works through the Django admin interface which lives at `/admin` of your application. Super users also have a link to “Site Admin” in the top bar.

The following settings affect the permissions of a user:

- **is_superuser** flag, users can do everything
- **groups**

- **Instrument Operator** – create flow cell, update flow cell created by the same user
- **Demultiplexing Operator** – create flow cell, update any flow cell
- **Demultiplexing Admin** – all of above, also CRUD of barcode set and instrument records
- **Import Bot** - create new flow cells, for future use

2.11.1 Notes on LDAP

Note that when using LDAP, your users first have to log into the Flowcelltool system to get their account created in the Flowcelltool database. You can then assign them into the appropriate roles.

When not using LDAP users, you can simply create users through the Django administration interface.

2.12 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

2.12.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/bihealth/flowcelltool/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

flowcelltool could always use more documentation, whether as part of the official flowcelltool docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/bihealth/flowcelltool/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

2.12.2 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.3, 3.4 and 3.5. Check https://travis-ci.org/bihealth/flowcelltool/pull_requests and make sure that the tests pass for all supported Python versions.

2.12.3 Tips

To run a subset of tests:

```
$ ./manage.py test
```

2.13 Credits

- Manuel Holtgrewe <manuel.holtgrewe@bihealth.de>
- Mikko Nieminen <mikko.nieminen@bihealth.de>

2.14 History

2.14.1 v0.3.0

- Adding UI for fast status updates.
- Refactoring concept of flowcell state, splitting into sequencing/conversion/delivery.
- Major refactoring of the UI and data models for automatization.
- Refactoring API and adding tests for it. Note that the API is still unstable (shown by having version v0), it will become properly versioned from v1 on.

2.14.2 v0.2.0

- Switching to UUID for all public-facing IDs.
- Adding support for message and multi-attachment upload via API.
- Adding basic profile page.
- Switching layout and vendoring JS/CSS dependencies.
- Adding adapter and quality JSON fields to FlowCell and APIs to set them.
- Some layout / UI refinements.
- Adding unstable API, mostly read-only except for what is needed for automated demultiplexing and QC.
- Adding UI for generating REST API login tokens.
- Allowing to specify message on login screen via environment variable.
- Adding version to stick footer.

2.14.3 v0.1.1

- Fixing display of libraries.

2.14.4 v0.1.0

- Initial release. Everything is new.

2.15 License

The MIT License (MIT)
Copyright (c) 2018, Berlin Institute of Health

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.