

请将手机调  
节至静音



支持环保  
支持胸牌再利用  
用后请投放至签到台  
纸箱内



illumina®

# BaseSpace® 基因云计算平台开发者训练营

## 零维度™ 简单成就未来

2018 年 2月 1日 上海



即插即用



轻松分享



安全可靠



无限拓展



# BaseSpace® 基因云计算平台开发者训练营

## 零维度™ 简单成就未来

2018 年 2月 1日 上海

13:00 - 13:30

BaseSpace Sequence Hub应用开发简介

13:30 - 14:30

BaseSpace Sequence Hub应用开发演示

14:30 - 14:45

茶歇

14:45 - 17:00

BaseSpace Sequence Hub应用开发练习

# BaseSpace Sequence Hub Development Conference

Paul Cherng, Product Manager

AAATCAACGTACCGTAACGAACGTATCATTAAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAAT  
GTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAAT  
GATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGCTTCTGT  
ATTACTTGATCCACTGATTCAACGTACCGTAAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTA  
GATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATTAACGTACCATTAAAGAGCTACCGTCTTCTGTAAACCT  
ACGAACGTATCAATTGAGACTAAATTAACGTACCATTAAAGAGCTACCGTGCAACGACGAAAAGAATGATAACAGTAACACACTTCTGTAAACCTTAAGAATCAACGTACCGTAACGAAC  
TTGATCCACTGATTCAACGTACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTG



# Overview of schedule

- **BSSH developer portal introduction**
- **BSSH app building demo with FastQC**
- **Coffee break**
- **Hands-on session, build your own FastQC app**
- **Closure ceremony**

# BSSH developer portal introduction



# BaseSpace Sequence Hub Native Apps

- **Enables push-button analysis with a graphical Input Form**
- **Applications can be written in any programming language and are executed within a Docker container**
- **Leverages AWS EC2 cloud infrastructure to run data analysis workloads**
- **Customizable report templates**

# Types of BaseSpace Sequence Hub Apps

- **Core apps**

- Developed by Illumina
- Fully supported

- **Lab apps**

- Developed by Illumina
- Limited support

- **3<sup>rd</sup> party apps**

- Developed externally
- Third-party developer responsible for support
- Can be public on app store or private





# Why build your own app?

- **Expose more options and settings for tools that might not be available in an existing app**
- **Deploy an existing analysis workflow to BaseSpace Sequence Hub to leverage the cloud computing infrastructure and enable push-button analysis for your lab**
- **Share your app with the community**

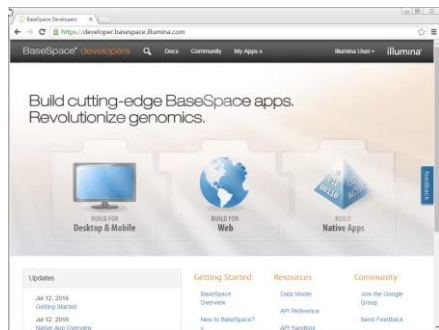
# App Development Overview

Set up new app in Developer Portal

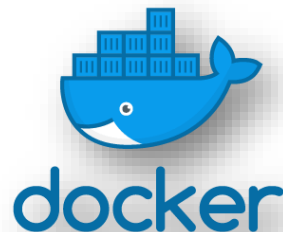
Create Input Form

Build Docker image

Customize Reports



The screenshot shows the 'FastQC v1.0.0' configuration form in BaseSpace Labs. The form is divided into several sections: 'Configuration' (with a logo), 'Output' (with 'App Session Name' set to 'FastQC 01/25/2018 1:05:48'), 'Output Project' (with a 'Select Project(s)' button), 'Input' (with 'Input Sample' and a 'Select Sample(s)' button), 'Options' (with 'Kmer Size' set to 5 and 'Use Contaminant Filter' checked), and a '1/4' indicator at the bottom.



```
class FASTQCApp(BaseSpaceNativeApp):  
    def __init__(self, input_directory:  
                  appsession_id=None):  
        BaseSpaceNativeApp.__init__(self,  
                                     workspace_directory=None)  
        self.input_directory = input_directory  
        self.appsession_id = appsession_id  
  
    def do_work(self, workspace_directory:  
                output_project = appsettings.  
                for sample in appsettings.input_directory:  
                    for fastq_set in sample.fastq_sets:  
                        #Analyze read 1 fastq
```



# AWS computing node configuration

- **Default standard instance type:**
  - 64-bit CPU with 32 virtual cores (16 physical cores)
  - 60.5 GiB memory
- **Other instance types available upon request**



## BSSH app building demo with FastQC

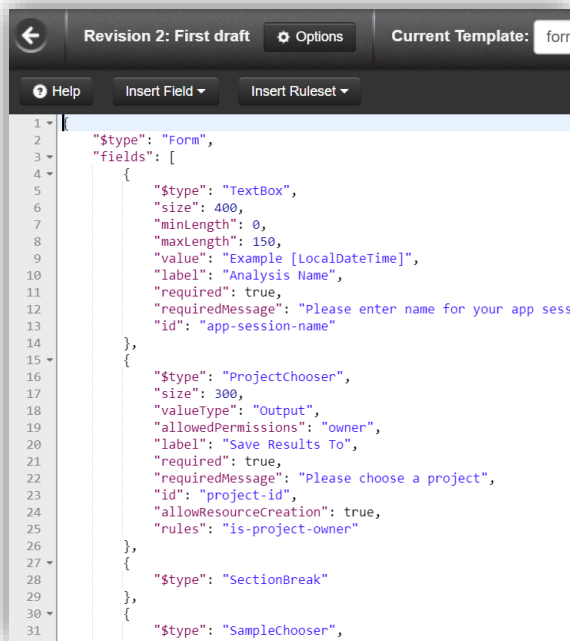


# Input Form Overview

- **The Input Form consists of two components:**
  - Form JSON
    - Describes which options and settings will be displayed
    - **(advanced)** Set up input validation rules
  - callbacks.js
    - Javascript code to configure how to launch the app by providing a Docker image and a Linux command to run
    - **(advanced)** configure how EC2 instances are created for parallel multi-node analysis
- **Documentation:**  
<https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/apptools/formbuilder-overview>

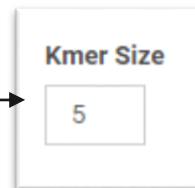
# How to build InputForm.json

- The “fields” property lists the input form elements

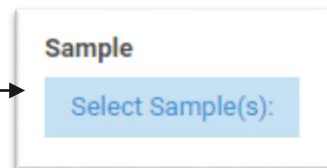


```
1- {
2-   "$type": "Form",
3-   "fields": [
4-     {
5-       "$type": "TextBox",
6-       "size": 400,
7-       "minLength": 0,
8-       "maxLength": 150,
9-       "value": "Example [LocalDateTime]",
10-      "label": "Analysis Name",
11-      "required": true,
12-      "requiredMessage": "Please enter name for your app sess",
13-      "id": "app-session-name"
14-    },
15-    {
16-      "$type": "ProjectChooser",
17-      "size": 300,
18-      "valueType": "Output",
19-      "allowedPermissions": "owner",
20-      "label": "Save Results To",
21-      "required": true,
22-      "requiredMessage": "Please choose a project",
23-      "id": "project-id",
24-      "allowResourceCreation": true,
25-      "rules": "is-project-owner"
26-    },
27-    {
28-      "$type": "SectionBreak"
29-    },
30-    {
31-      "$type": "SampleChooser",
```

```
{
  "$type": "Numeric",
  "id": "kmer-size",
  "size": 50,
  "required": true,
  "label": "Kmer Size",
  "min": 2,
  "max": 10,
  "value": 5,
  "numericType": "Integer"
}
```



```
{
  "$type": "SampleChooser",
  "size": 300,
  "valueType": "Input",
  "allowedPermissions": "read",
  "label": "Sample",
  "required": false,
  "id": "sample-id",
  "rules": "sample-reader"
}
```



# How to write callback.js

- **launchSpec(dataProvider) function defines how the app will be launched**
  - At minimum, launchSpec(dataProvider) must return an object with the following properties:
    - **containerImageId** (type: str):
      - The name of the Docker image containing the data analysis workflow
      - Example: docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app
    - **commandLine** (type: [str]):
      - A string array that specifies the command to be executed within the Docker container
      - Example: [ "/usr/bin/python", "/fastqc-app.py", "/data/input", "/data/output", "/data/scratch", "/data/logs" ]
    - **Options** (type: [str]):
      - A string array containing a list of BaseSpace Native App Engine options.
      - Example: [ "bsfs.enabled=true" ]
      - **Note: You should not modify this setting**

# How to write callback.js

- **Example callbacks.js script for FASTQC App**

```
function launchSpec(dataProvider)
{
    var ret = {
        containerImageId: "docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app",
        commandLine: [ "/usr/bin/python",
            "/fastqc-app.py",
            "/data/input",
            "/data/output",
            "/data/scratch",
            "/data/logs"
        ],
        Options: [ "bsfs.enabled=true" ]
    };
    return ret;
}
```



# What is spacedock?

- **Service running on EC2 instance**

- Downloads Docker image and input files
- Starts Docker container with commandLine specified in callbacks.js
- Uploads files and AppResults back to BaseSpace Sequence Hub after analysis finishes

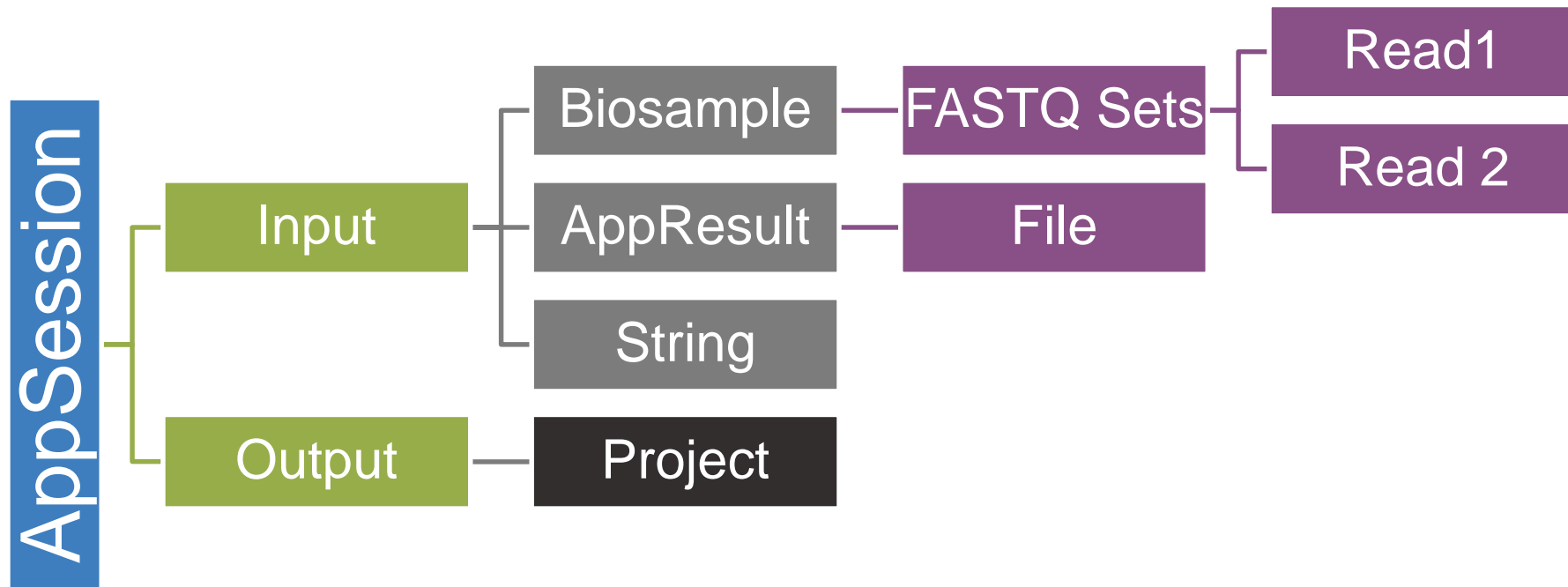
- **Input file download locations**

- Samples: /data/inputs/samples/<SampleID>
- AppResults and Files: /data/inputs/appresults/<AppResultID>

- **Output file locations**

- Each AppResult should be written to /data/outputs/<ProjectID>/<AppResultName>

# BaseSpace Sequence Hub App Development Data Model



# Appsession.json example and interpretation

- When the app launches, spacedock will write an “AppSession.json” file to /data/inputs/AppSession.json
- This file contains all the user-selected inputs, settings, and options from the Input Form
- Properties can be parsed using a JSON deserializer library from any language

# AppSession.json

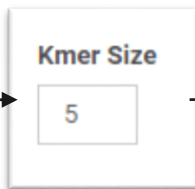
## Example Input.kmer-content property

Input Form JSON

User Selection

AppSession.json property

```
{
  "$type": "Numeric",
  "valueType": "Input",
  "id": "kmer-size",
  "size": 50,
  "required": true,
  "label": "Kmer Size",
  "min": 2,
  "max": 10,
  "value": 5,
  "numericType": "Integer"
}
```



Kmer Size

5

```
{
  "Type": "string",
  "Href": "v1pre3/appsessions/75467504/properties/Input.kmer-size",
  "Name": "Input.kmer-size",
  "Description": "Kmer Size",
  "Content": "5"
},
```

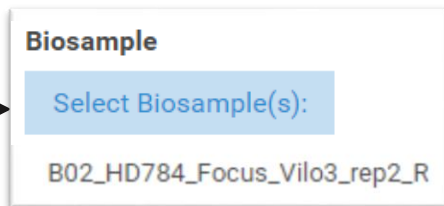
# AppSession.json

## Example Input.sample-id property

### Input Form JSON

```
{
  "$type": "SampleChooser",
  "size": 300,
  "valueType": "Input",
  "allowedPermissions": "read",
  "label": "Sample",
  "required": false,
  "id": "sample-id",
  "rules": "sample-reader"
}
```

### User Selection

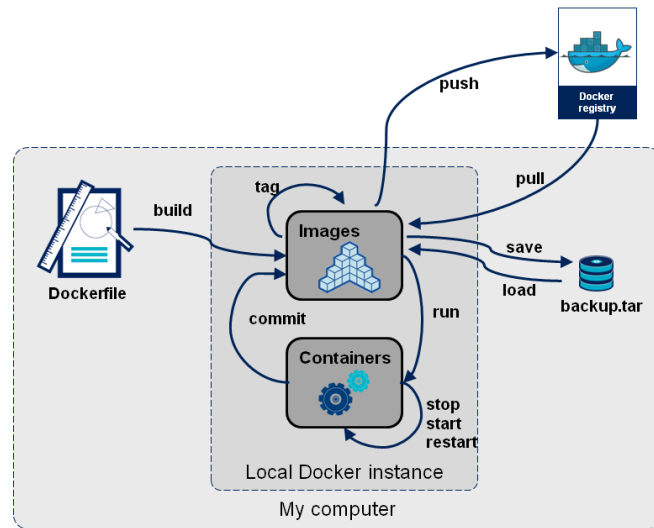


### AppSession.json property

```
{
  "Type": "sample",
  "Href":
    "v1pre3/appsessions/75467504/properties/Input.sample-id",
  "Name": "Input.sample-id",
  "Description": "Biosample",
  "Content": {
    "Id": "60332289",
    "Href": "v1pre3/samples/60332289",
    "IsMerged": false,
    "UserOwnedBy": {
      "Id": "7016",
      "Href": "v1pre3/users/7016",
      "Name": "Eric Allen",
      "DateCreated": "0001-01-01T00:00:00.0000000",
    },
  },
  "Name": "B02_HD784_Focus_Vilo3_rep2_R",
  "SampleId": "B02_HD784_Focus_Vilo3_rep2_R",
  "Status": "Complete",
  "StatusSummary": "",
  "DateCreated": "2017-10-25T23:36:02.0000000",
  "TotalSize": 129967276,
  "Read1": 151,
  "Read2": 151,
  "NumReadsPF": 1000000,
  "NumReadsRaw": 1000000,
  "TotalReadsRaw": 2000000,
  "TotalReadsPF": 2000000,
  "TotalClustersRaw": 1000000,
  "TotalClustersPF": 1000000,
  "IsPairedEnd": true
},
}
```

# Docker common usage

- A Docker image is a package that includes an application and all of its dependencies, including the operating system
- Dockerfile is a text file that contains instructions telling the Docker software how to build a Docker image
  - Example command:
    - `docker build -t <image name> -f <path to Dockerfile>`
- Docker images need to be uploaded to a Docker registry before they can be used in a BaseSpace App
  - Example command:
    - `docker push <image name>`
- Documentation:
  - <https://docs.docker.com/>
  - <https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/native-apps/manage-docker-image>



# FASTQC Docker Image Schematic

Docker Image

CentOS Operating System

Python 2.7

Java JDK

FASTQC

FASTQC  
BaseSpace App  
Python Script

# How to write app wrapper script using Python library

- **bssh\_native\_app Python (2.7) library assists with:**
  - Parsing and extracting inputs from AppSession.json properties
  - Uploading files and AppResults to BaseSpace Sequence Hub
- **Basic usage**
  1. Import bssh\_native\_app Python modules
  2. Inherit from ***BaseSpaceNativeApp*** class
  3. Override ***do\_work(self, workspace\_directory, appsettings, output\_builder)*** class method
    1. Parse AppSession.json properties using ***appsettings*** object
    2. Perform data analysis on input data
    3. Create AppResults for upload back to BaseSpace Sequence Hub



# Example FASTQC App Python wrapper

## **#import modules**

```
import subprocess
import os
import sys
from bssh_native_app.BaseSpaceNativeApp import BaseSpaceNativeApp
```

## **#inherit BaseSpaceNativeApp class**

```
class FASTQCApp(BaseSpaceNativeApp):
```

### **#constructor**

```
def __init__(self, input_directory_path, output_directory_path, scratch_directory_path, log_directory_path, appsession_id=None):
    #use default BaseSpaceNative App constructor
    BaseSpaceNativeApp.__init__(self, input_directory_path, output_directory_path, scratch_directory_path,
                                log_directory_path, appsession_id)
```

# How to write app wrapper script using Python library

```
class BaseSpaceNativeApp
```

```
    def do_work(self, workspace_directory, appsettings, output_builder):
```

- **do\_work** class method contains all app execution logic
  - Parameters:
    - **workspace\_directory** (type: string) – the path to the scratch space directory where all temporary files should be written to
    - **appsettings** (type: AppSettings) – object containing all the Input and Output properties from AppSession.json
    - **output\_builder** (type: OutputBuilder) – object used to assist with creating AppResults to be uploaded back to BaseSpace Sequence Hub

# How to write app wrapper script using Python library

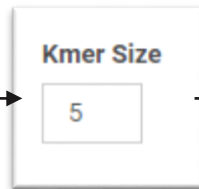
class AppSettings

- Used to parse and extract user-selected options and settings from Input Form

Input Form JSON

```
{
  "$type": "Numeric",
  "valueType": "Input",
  "id": "kmer-size",
  "size": 50,
  "required": true,
  "label": "Kmer Size",
  "min": 2,
  "max": 10,
  "value": 5,
  "numericType": "Integer"
}
```

User Selection



AppSession.json property

```
{
  "Type": "string",
  "Href": "v1pre3/appsessions/75467504/properties/Input.kmer-size",
  "Name": "Input.kmer-size",
  "Description": "Kmer Size",
  "Content": "5"
},
```

Python script

```
def do_work(self, workspace_directory, appsettings, output_builder):
    kmer_size = appsettings.inputs["kmer-size"].content[0]
```

# How to write app wrapper script using Python library

class OutputBuilder

- **Used to create AppResults and select files to be uploaded to BaseSpace Sequence Hub**
- **Examples:**

```
#create an AppResult
```

```
output_appresult = output_builder.create_output_appresult(output_project, "AppResult Name", "AppResult Description")
```

```
#add a file to the AppResult
```

```
output_appresult.add_file("/path/to/file.txt")
```

```
#add a directory to the AppResult and rename it to "Files"
```

```
output_appresult.add_directory("/path/to/directory/", "Files")
```

# Example FASTQC App Python wrapper

```
class FASTQCApp(BaseSpaceNativeApp):
    def __init__(self, input_directory_path, output_directory_path, scratch_directory_path, log_directory_path, appsession_id=None):
        BaseSpaceNativeApp.__init__(self, input_directory_path, output_directory_path, scratch_directory_path,
                                     log_directory_path, appsession_id)

    #override do_work class method
    def do_work(self, workspace_directory, appsettings, output_builder):
        # Parse AppSession.json properties using appsettings object
        output_project = appsettings.input["project-id"].content[0]
        sample = appsettings.input["sample-id"].content[0]

        for fastq_set in sample.fastq_sets:
            # Perform data analysis with FASTQC tool on read 1 FASTQ file
            read_1_fastq_filename = os.path.basename(fastq_set.read_1_file_path)
            exit_code, output_directory_path_1 = run_fastqc(fastq_set.read_1_file_path, workspace_directory)

            if exit_code == 0:
                # Create AppResults for upload back to BaseSpace Sequence Hub
                output_appresult_1 = output_builder.create_output_appresult(output_project,
                                                                              read_1_fastq_filename, "Results for " + read_1_fastq_filename)
                output_appresult_1.add_directory(output_directory_path_1, "Files")
```

# Example FASTQC App Python wrapper

```
def run_fastqc(fastq_file_path, workspace_directory):
    def get_unique_directory_path(candidate_directory_path):
        count = 0
        directory_path = candidate_directory_path
        while os.path.isdir(directory_path):
            count = count + 1
            directory_path = os.path.dirname(candidate_directory_path) + "_" + str(count) + "/"
        return directory_path

    candidate_output_directory = os.path.join(workspace_directory, os.path.basename(fastq_file_path) + "/")
    output_directory_path = get_unique_directory_path(candidate_output_directory)
    os.makedirs(output_directory_path)
    #run FASTQC
    fastqc_command = ["fastqc", "--outdir", output_directory_path, "--extract", fastq_file_path]
    exit_code = subprocess.call(fastqc_command)

    return exit_code, output_directory_path
```

- Documentation for subprocess class:  
<https://docs.python.org/2/library/subprocess.html>

# Example FASTQC App Python wrapper

## #App Python script entrypoint

```
if __name__ == "__main__":  
    input_dir = sys.argv[1]  
    output_dir = sys.argv[2]  
    scratch_dir = sys.argv[3]  
    logs_dir = sys.argv[4]  
  
    app = FASTQCApp(input_dir, output_dir, scratch_dir, logs_dir)  
  
    app.start()
```

## //Input Form callbacks.js

```
function launchSpec(dataProvider)  
{  
    var ret = {  
        containerImageId: "pcherngilmn/fastqc-app",  
        commandLine: [ "/usr/bin/python",  
                        "/fastqc-app.py",  
                        "/data/input",  
                        "/data/output",  
                        "/data/scratch",  
                        "/data/logs"  
                    ],  
        Options: [ "bsfs.enabled=true" ]  
    };  
    return ret;  
}
```

# How to build report

- Reports templates are built using HTML, CSS, Javascript,
- Liquid template language enables accessing AppResult files during report generation, as well as some scripting capabilities
- Documentation available:  
<https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/apptools/reportbuilder-overview>
- Example:  

```
<!-- find URL of "error.png" file using Liquid template language-->
{% assign img_error = result.files.where.ends_with("error.png").first.href %}
<!-- display "error.png" in the report using HTML -->
<img src={{img_error}}>
```



# Example FASTQC Report Template

```
{% assign txt_summary = result.files.where.ends_with["summary.txt"].first %}
{% assign img_tick = result.files.where.ends_with["tick.png"].first.href %}
{% assign img_warning = result.files.where.ends_with["warning.png"].first.href %}
{% assign img_error = result.files.where.ends_with["error.png"].first.href %}
{% assign img_per_base_quality = result.files.where.ends_with["per_base_quality.png"].first.href %}
{% assign status_per_base_quality = txt_summary.tsv.parse.to_array[0][0] | stringify %}
```

```
<!doctype html>
<html>
<head>
  <title>{{ result.name }}</title>
  <link rel="stylesheet" type="text/css"
    href="https://da1s119xsxmu0.cloudfront.net/libraries/bootstrap/1.0.0/css/master.min.css" />
</head>
<body>
  <div class="row-fluid">
    <div class="col-xs-12">
      <h3>{{ result.name }}</h3>
      <div id="per-base-quality">
        {% assign section_header = "Per base sequence quality" %}
        {% case status_per_base_quality %}
          {% when "PASS" %}
            <h4>{{section_header}}</h4>
          {% when "WARN" %}
            <h4>{{section_header}}</h4>
          {% when "FAIL" %}
            <h4>{{section_header}}</h4>
        {% endcase %}
        
      </div>
    </div>
  </div>
</body>
</html>
```

# Demo – Actual launch of the app and show output

- **FASTQC demo input form:**
  - <https://cn1.sh.basespace.illumina.com.cn/apps/42042/start>
- **FASTQC demo output:**
  - <https://cn1.sh.basespace.illumina.com.cn/analyses/242244>

## Coffee break



## Hands-on session, build your own FastQC app



# Brief introduction to files needed (handed out in flash)

- **03\_Input\_Form**

- input\_form.json
- callbacks.js

- **04\_Docker\_Image**

- Dockerfile
- bssh\_native\_app-0.9.1.0.tar.gz
- fastqc-app.py
- simple\_fastqc.tar

- **05\_Report\_Builder**

- report.html

# How to log into our Linux server

- **Windows:**

- Install Git-2.16.1-64-bit.exe from “01\_SSH\_Client\_Installer” folder in flash drive
- Run Git Bash

- **Mac/Linux:**

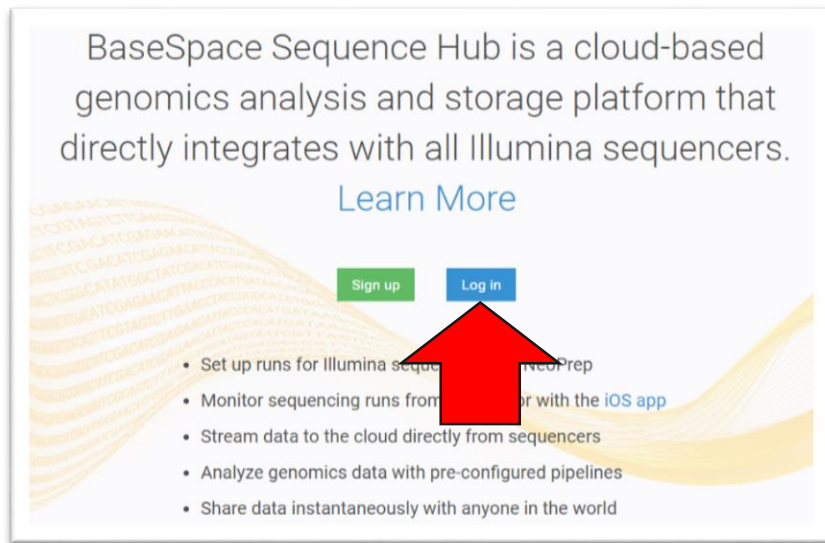
- Use the terminal

# How to log into our Linux server

- **Put your key file <control.data\_upload> at current path**
- **Modify read-write permission setting**
  - `chmod 700 control.data_upload`
- **Login with this command**
  - `ssh -i control.data_upload <username>@<ip address>`
  - Example: [testuser@123.123.123.123](#)
  - Username and IP address will be provided to you on pieces of paper
- **Input phase for key file: LMiuOpDZqXzOB5Py1J**
- **Create your working directory under “/data”, for example**
  - `mkdir /data/<username>`

# Log into BaseSpace Sequence Hub

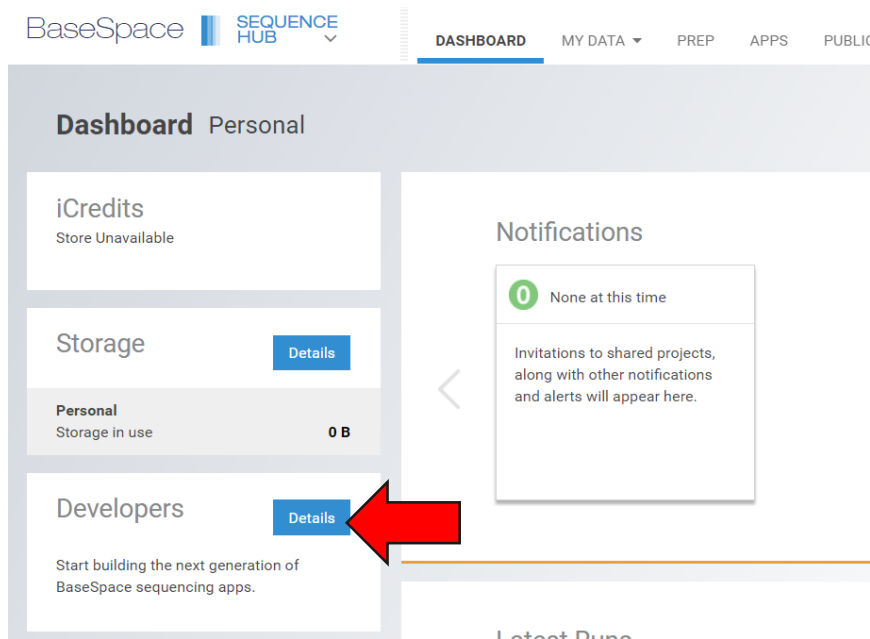
- Go to <https://cnn1.sh.basespace.illumina.com.cn>
- Log in or create a new account





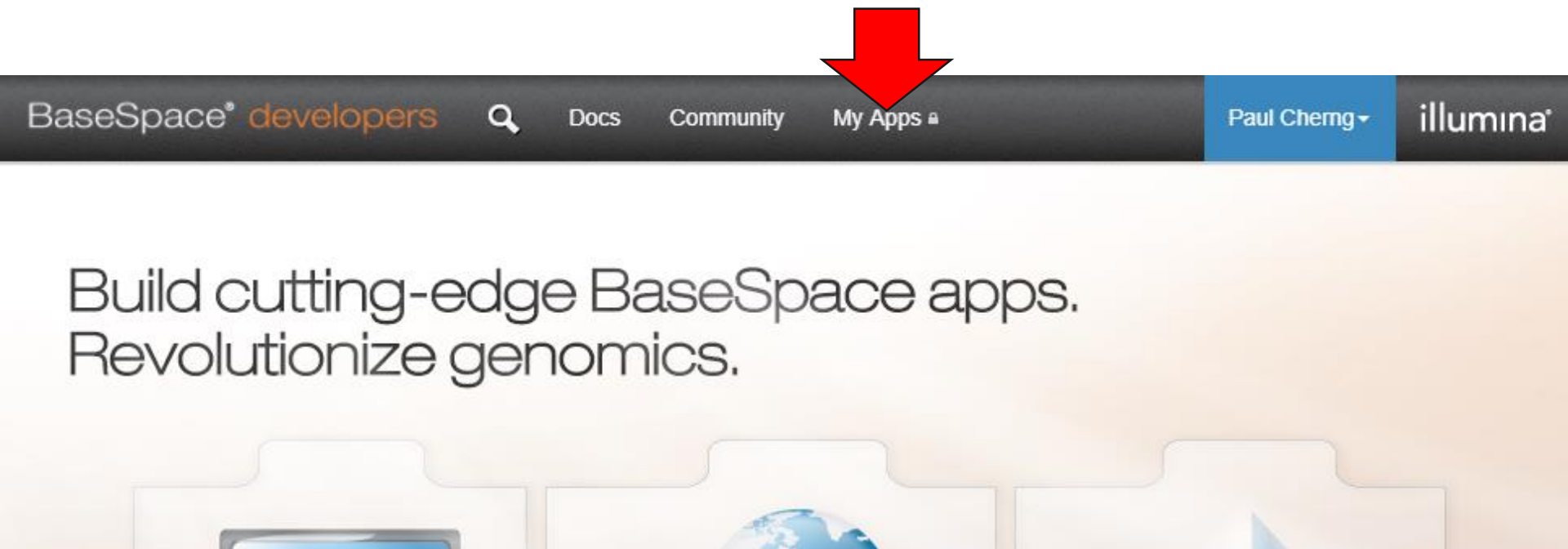
# Access Developer Portal

- Click on Developers “Details” button on the left



# Access Developer Portal

- From the main toolbar, select My Apps



# Access Developer Portal

- **Accept the Developer Portal Agreement**

- Only need to do this once

## g. Change in Terms

Illumina reserves the right to change these terms at any time in its discretion. When we make changes, shall become effective on the first day of third calendar following the date such notice was sent (e.g., if t published Application represents your agreement to any updated terms.

## h. Force Majeure

You agree that Illumina is not responsible to you for any failure to perform or delay attributable in whole government actions, actual or threatened acts of war, terrorism, civil disturbance or insurrection, sabotage materials or equipment.

## i. Severability

If any provision of this Agreement is held invalid or unenforceable, such provision shall be enforced to th

## j. Notices

All notices and claims by you regarding Sequence Hub, the API, your Application, this Agreement, and t

## k. Entire Agreement

These terms represent the entire agreement between you and Illumina regarding the subject matter her Illumina's right to revise these terms, these terms may only be amended or waived by a written agreeme

September 15, 2017

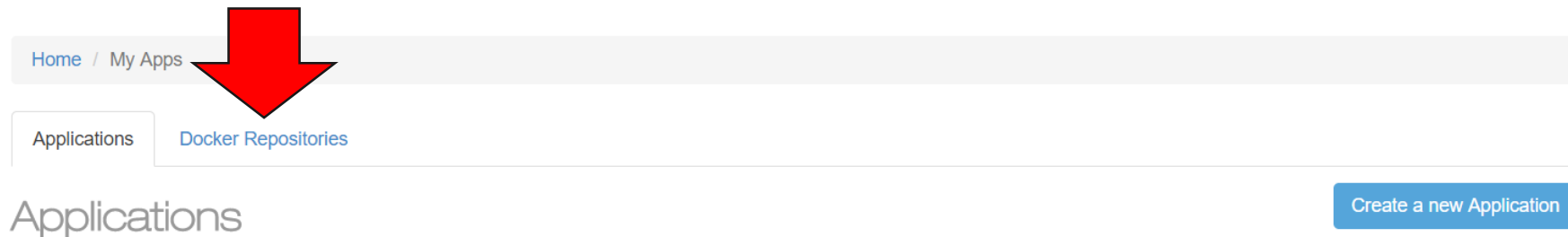


I Accept The Agreement

Cancel

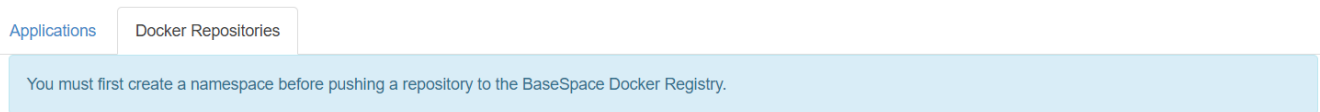
# Create new Docker repository namespace

- Click on “Docker Repositories” tab



# Create new Docker repository namespace

- Enter a namespace in the textbox and click “Create Namespace”
  - Only need to do this once
  - Remember your namespace because you will need it later

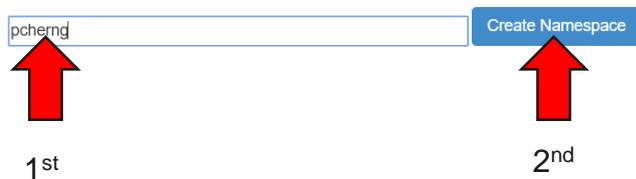


## What is a namespace?

A namespace is a uniquely named container for your repositories. Only you will have access to the repositories in your namespace.

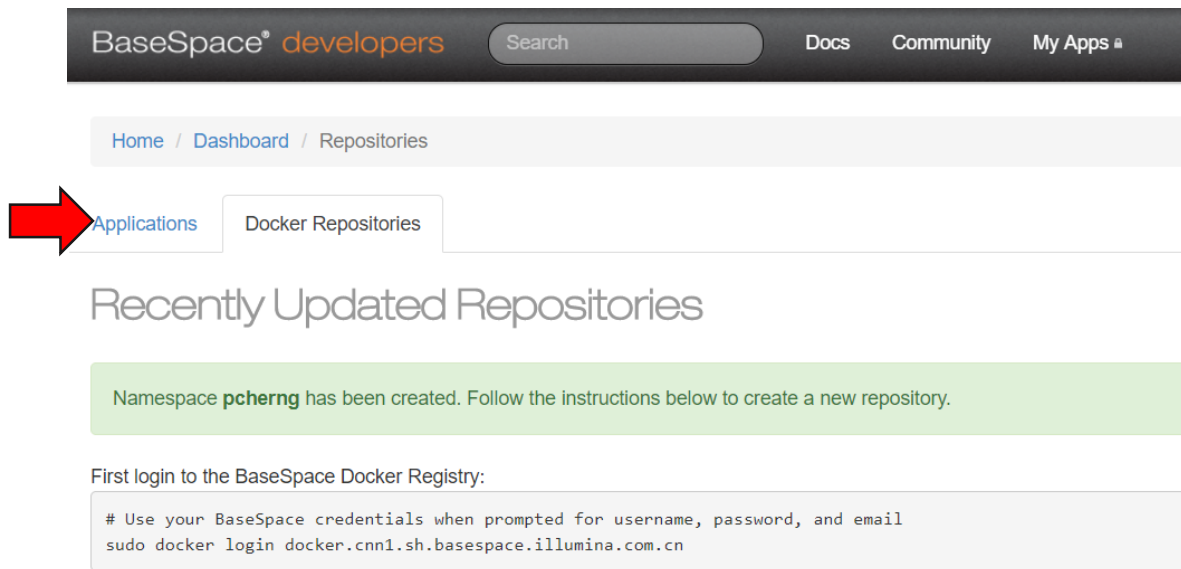
## What name should I choose?

A namespace is like a username. You may want to use your first initial and last name or the name of your organization or product. You will be able to create only one namespace. Once a namespace has been created it cannot be renamed.



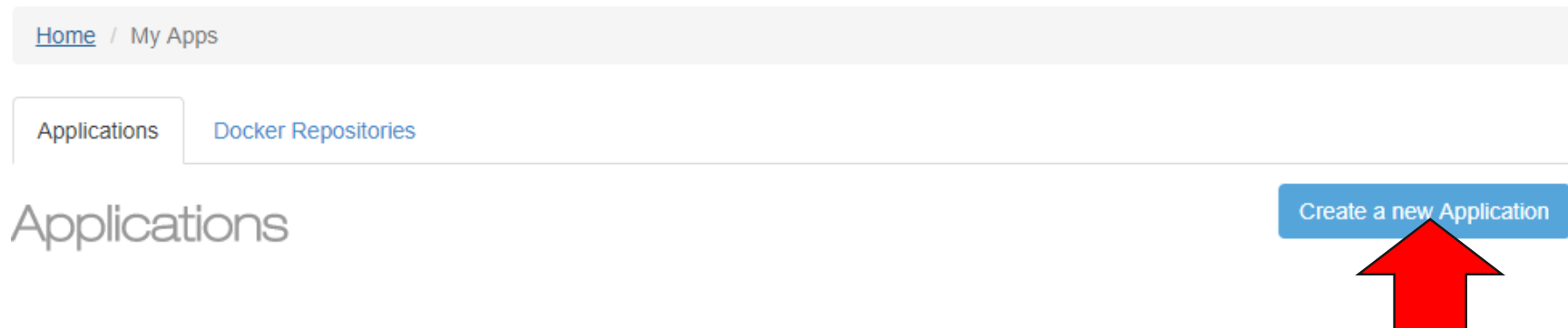
# Creating an Application in BaseSpace Sequence Hub

- Click on the “Applications” tab



# Creating an Application in BaseSpace Sequence Hub

- Click on “Create a new Application”



# Creating an Application in BaseSpace Sequence Hub

- Fill in the Application Details and click “Create Application”

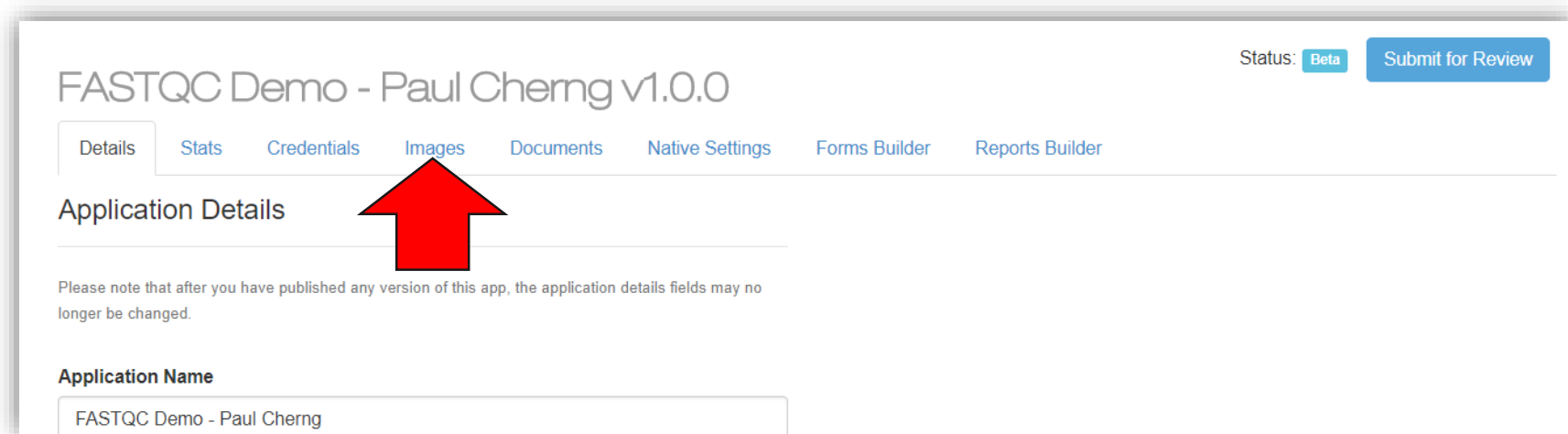
The screenshot shows the 'Create an application' page in the BaseSpace developers interface. The page has a dark header with the 'BaseSpace\* developers' logo, a search bar, and navigation links for 'Docs', 'Community', and 'My Apps'. The user 'Paul Cherng' is logged in, and the 'illumina' logo is in the top right. Below the header is a breadcrumb trail: 'Home / My Applications / Create an application'. The main section is titled 'Application Details' and contains several form fields:

- Name:** A text input field containing 'FASTQC Demo - Paul Cherng'. A red arrow points to this field.
- Company or Organization Name:** A text input field containing 'Illumina, Inc.'. A red arrow points to this field.
- App Type:** A dropdown menu with 'Native' selected.
- Short Description:** A text area containing the text: 'FASTQC is used for quality control checks on raw sequence data coming from high throughput sequencing pipelines'. A red arrow points to this field.
- Create Application:** A blue button with the text 'Create Application'. A red arrow points to this button.
- Cancel:** A light blue button with the text 'Cancel'.



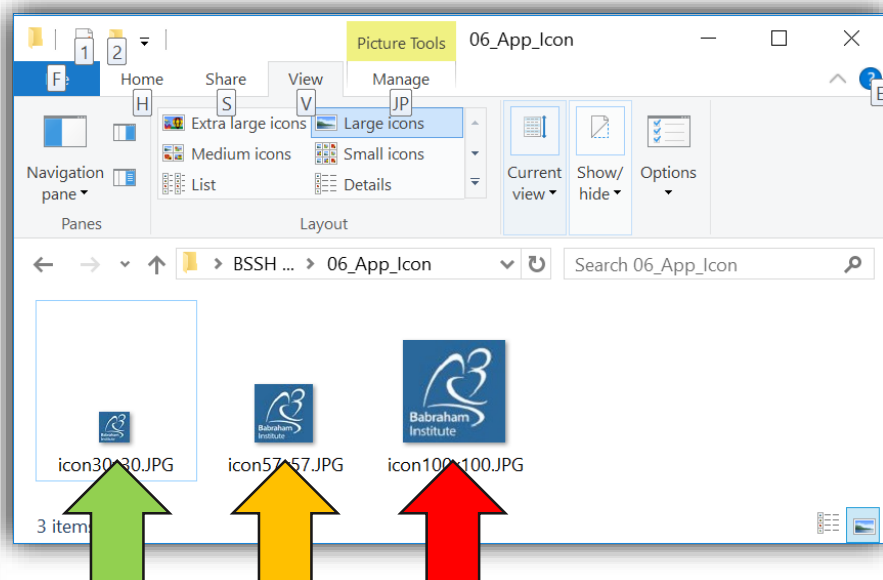
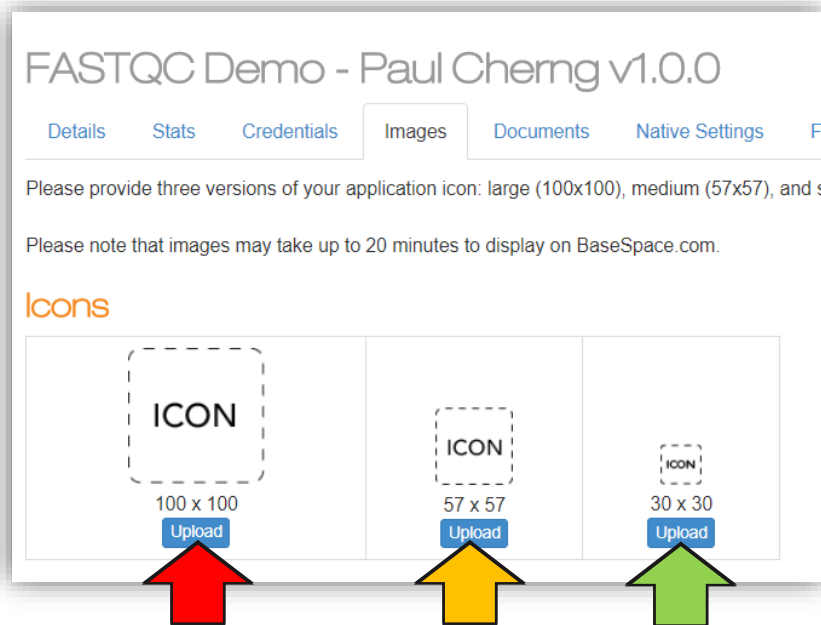
# Add App Icons

- Click on “Images” tab



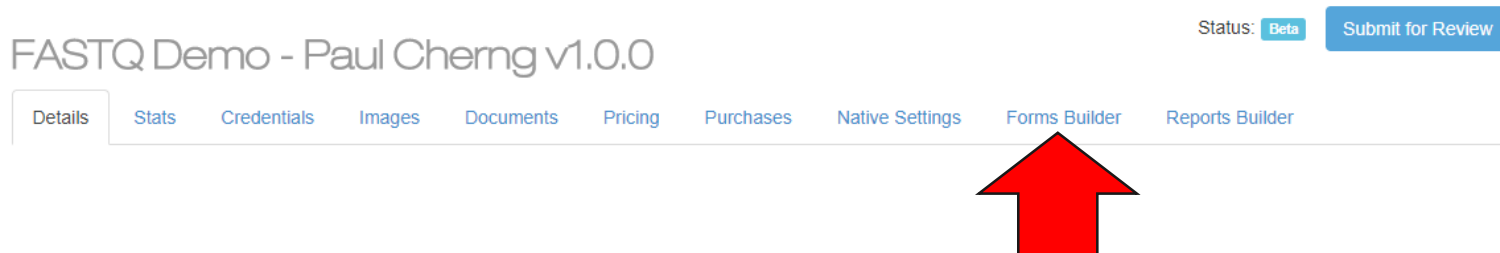
# Add App Icons

- Upload the icon images in the “06\_App\_Icon” folder of the flash drive



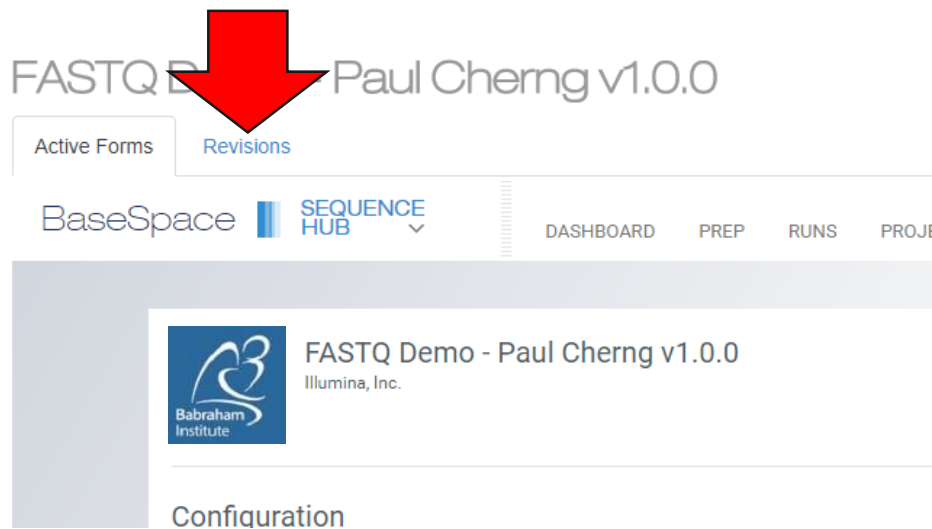
# Set up Input Form

- On Details page, click on “Forms Builder”



# Set up Input Form

- On Forms Builder page, click on “Revisions” tab



# Set up Input Form


- On Revisions tab, select version 1 of the input form and then click “Duplicate”



FASTQ Demo - Paul Cherng v1.0.0

Status: Beta Submit for Review

Revisions

☒ Use a custom form (below). If this option is unchecked, a generic default form will be shown to users instead.

2nd  + Duplicate 🗑 Delete

	Version	Description	Publish Status	DateModified
1st 	 1	Example	<span>Active</span>	5 minutes ago

# Set up Input Form

- After making a copy of version 1, click on “Copy of version 1”

FASTQ Demo - Paul Cherng v1.0.0

Status: Beta Submit for Review

Revisions

☒ Use a custom form (below). If this option is unchecked, a generic default form will be shown to users instead.

+ Duplicate 🗑 Delete

Version	Description	Publish Status	DateModified
2	Copy of version 1	Inactive	in a few seconds
1	Example	Active	10 minutes ago

# Set up Input Form

- Copy and paste “input\_form.json” content from “03\_Input\_Form” folder in the flash drive to the Input Form editor and then click “Save and Preview”

1<sup>st</sup>  
(Copy and paste)

2<sup>nd</sup>

```
{
  "$type": "Form",
  "fields": [
    {
      "$type": "Textbox",
      "size": 400,
      "minLength": 0,
      "maxLength": 150,
      "value": "Example [LocalDateTime]",
      "label": "Analysis Name",
      "required": true,
      "requiredMessage": "Please enter name for your app session.",
      "id": "app-session-name"
    },
    {
      "$type": "ProjectChooser",
      "size": 300,
      "valueType": "Output",
      "allowedPermissions": "owner",
      "label": "Save Results To",
      "required": true,
      "requiredMessage": "Please choose a project",
      "id": "project-id",
      "allowResourceCreation": true,
      "rules": "is-project-owner"
    },
    {
      "$type": "SectionBreak"
    },
    {
      "$type": "SampleChooser",
      "size": 300,
      "valueType": "Input",
      "allowedPermissions": "read",
      "label": "Sample",
      "required": false,
      "id": "sample-id",
      "rules": "sample-reader"
    },
    {
      "$type": "Numeric",
      "id": "kmer-size"
    }
  ]
}
```

# Set up Input Form

1. Switch to callbacks.js template
2. Copy and paste “callbacks.js” content from “03\_Input\_Form” folder in the flash drive to the Input Form editor
3. Replace “<your docker image name>” with a Docker image name
  - **Format:** docker.cnn1.sh.basespace.illumina.com.cn/<docker namespace>/<repo name>
  - **Example:** docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app
4. Click “Save and Preview”

1<sup>st</sup>

4<sup>th</sup>

2<sup>nd</sup>  
(Copy and paste)

3<sup>rd</sup>  
(replace “<your docker image name>”)

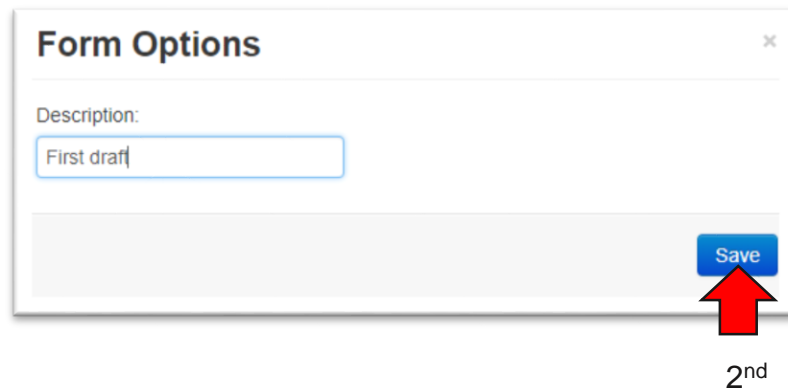
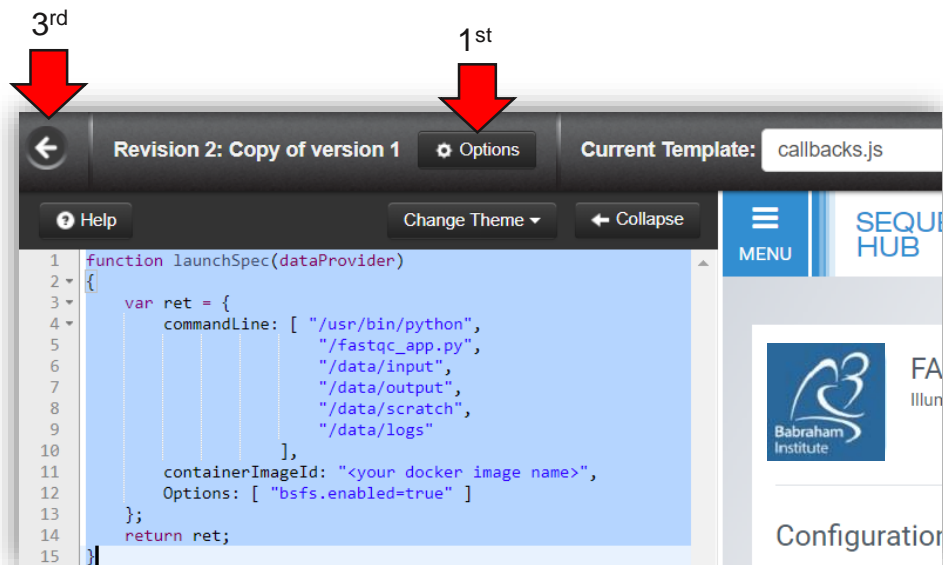
```
function launchSpec(dataProvider)
{
  var ret = {
    commandLine: [ "/usr/bin/python",
                  "/fastqc_app.py",
                  "/data/input",
                  "/data/output",
                  "/data/scratch",
                  "/data/logs"
                ],
    containerImageId: "docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app",
    Options: [ "bsfs.enabled=true" ]
  };
  return ret;
}
```

For Research Use Only. Not for use in diagnostic procedures.



# Set up Input Form

- Click on “Options” and give the revision a new name, then return to the Revisions page by clicking the “←” button



# Set up Input Form


- Select the newly created Input Form revision, and then click “Activate”.
- The Input Form is now complete.

FASTQ Demo - Paul Cherng v1.0.0

Status: Beta Submit for Review

Revisions

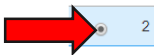
☒ Use a custom form (below). If this option is unchecked, a generic default form will be shown to users instead.

2nd 

Activate

Duplicate

Delete

	Version	Description	Publish Status	DateModified
1st 	2	First draft	Inactive	a few seconds ago
	1	Example	Active	37 minutes ago

# Tag Docker Image

- Run docker images to see existing Docker images on the EC2 instance

```
[pcherng@ip-10-255-130-200 ~]$ docker images
```

REPOSITORY	IMAGE ID	CREATED	SIZE	TAG
docker.cnn1.sh.basespace.illumina.com.cn/jmx/simple_fastqc	6ac7a29e6e49	5 days ago	807MB	latest
docker.cnn1.sh.basespace.illumina.com.cn/luohx/fastqc	6ac7a29e6e49	5 days ago	807MB	latest
docker.cnn1.sh.basespace.illumina.com.cn/niu_test/simple_fastqc	6ac7a29e6e49	5 days ago	807MB	latest
docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app	6ac7a29e6e49	5 days ago	807MB	latest
docker.cnn1.sh.basespace.illumina.com.cn/christie/fastqc	6ac7a29e6e49	5 days ago	807MB	latest
docker.cnn1.sh.basespace.illumina.com.cn/jjxu/simple_fastqc	6ac7a29e6e49	5 days ago	807MB	latest
docker.illumina.com/jjxu/simple_fastqc	063ea5ef91a2	5 days ago	807MB	latest
ubuntu	2a4cca5ac898	2 weeks ago	111MB	latest

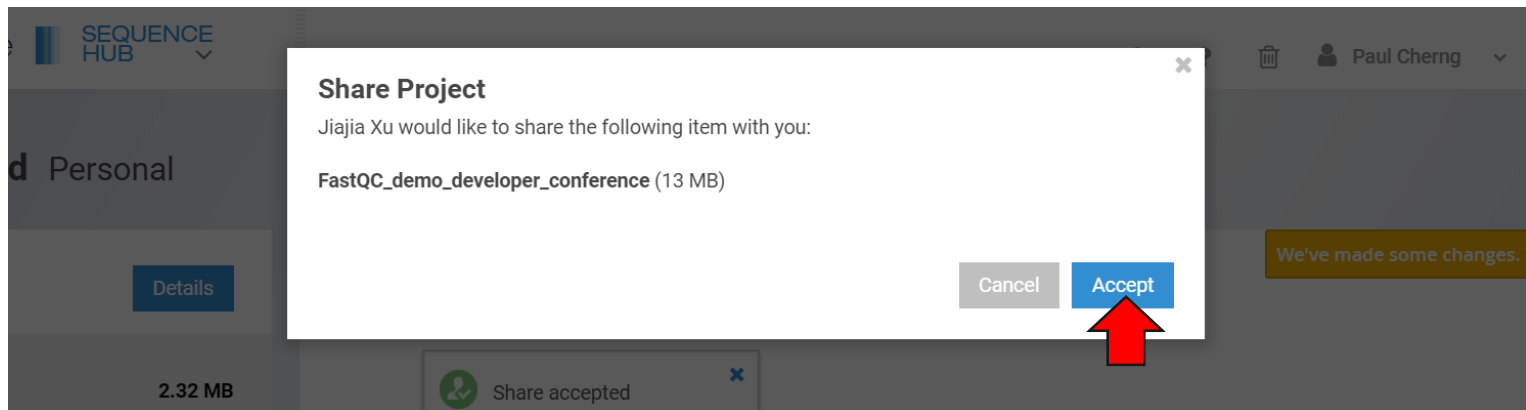
- Tag the simple\_fastqc image with a new name with this command:
  - docker tag docker.cnn1.sh.basespace.illumina.com.cn/jjxu/simple\_fastqc  
docker.cnn1.sh.basespace.illumina.com.cn/<docker namespace>/fastqc-app

# Push Docker Image to registry

- **Push the Docker image to the Illumina Docker registry with this command:**
  - `docker push <docker image name>`
  - **<docker image name>** should be replaced with whatever you entered for containerId in the Input Form callbacks.js script
  - Example:
    - `docker push docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app`

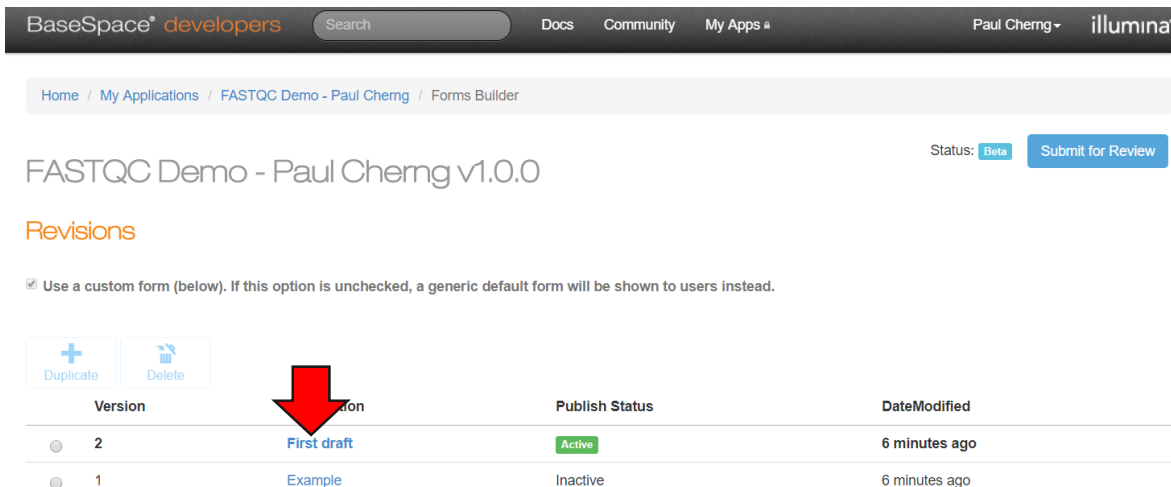
# Test the app

- **Accept the project share for the demo dataset:**
  - <https://cn1.sh.basespace.illumina.com.cn/s/NIXKZnezr0o7>



# Test the app

- Return to the Input Form “Revisions” tab
- Click on the activated Input Form revision



BaseSpace<sup>®</sup> developers Search Docs Community My Apps ▾ Paul Cherng ▾ illumina<sup>®</sup>

Home / My Applications / FASTQC Demo - Paul Cherng / Forms Builder

FASTQC Demo - Paul Cherng v1.0.0 Status: Beta Submit for Review

Revisions

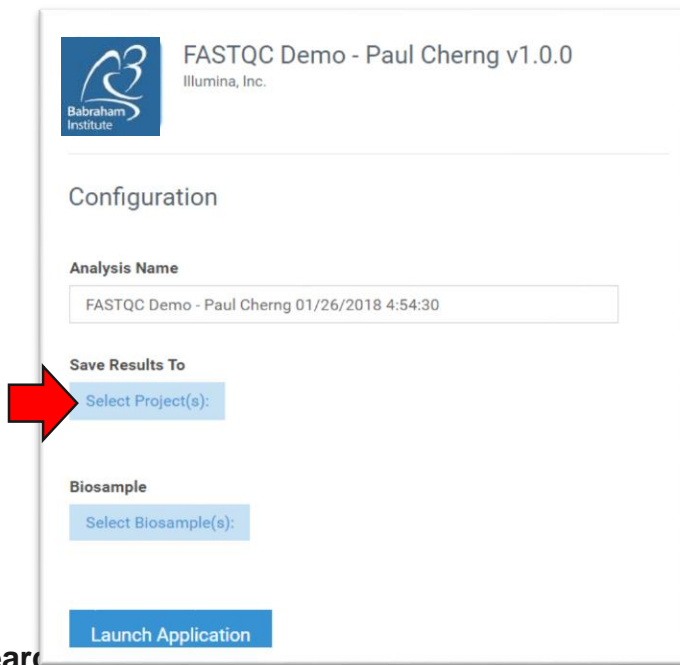
☒ Use a custom form (below). If this option is unchecked, a generic default form will be shown to users instead.

+ Duplicate 🗑 Delete

	Version	Revision	Publish Status	Date Modified
<input checked="" type="radio"/>	2	<a href="#">First draft</a>	<span>Active</span>	6 minutes ago
<input type="radio"/>	1	<a href="#">Example</a>	Inactive	6 minutes ago

# Test the app

- In the Input Form preview, click on “Select Project(s)”



The screenshot shows the FASTQC Demo application interface. At the top left is the Babraham Institute logo. To its right, the text reads "FASTQC Demo - Paul Cherng v1.0.0" and "Illumina, Inc.". Below this is a "Configuration" section. It contains an "Analysis Name" field with the text "FASTQC Demo - Paul Cherng 01/26/2018 4:54:30". Underneath is a "Save Results To" section with a blue button labeled "Select Project(s):". A large red arrow points to this button. Below that is a "Biosample" section with a blue button labeled "Select Biosample(s):". At the bottom is a blue button labeled "Launch Application".

# Test the app

- Create a new Project called “FASTQC Test”

1<sup>st</sup>

A screenshot of the 'Select Project(s)' dialog. It has a title bar with a close button. Below the title is a search bar labeled 'Search...'. A status bar says 'Showing 1 of 1'. Below that is a section titled 'PROJECTS' with a help icon. The main area is empty. At the bottom are 'Cancel' and 'Select' buttons. A red arrow points to the 'New' button at the bottom left.

2<sup>nd</sup>

A screenshot of the 'New Project' dialog. It has a title bar with a close button. Below the title is a 'Name' field containing 'FASTQC Test'. Below that is a 'Description' field. At the bottom right are 'Cancel' and 'Create' buttons. A red arrow points to the 'Name' field.

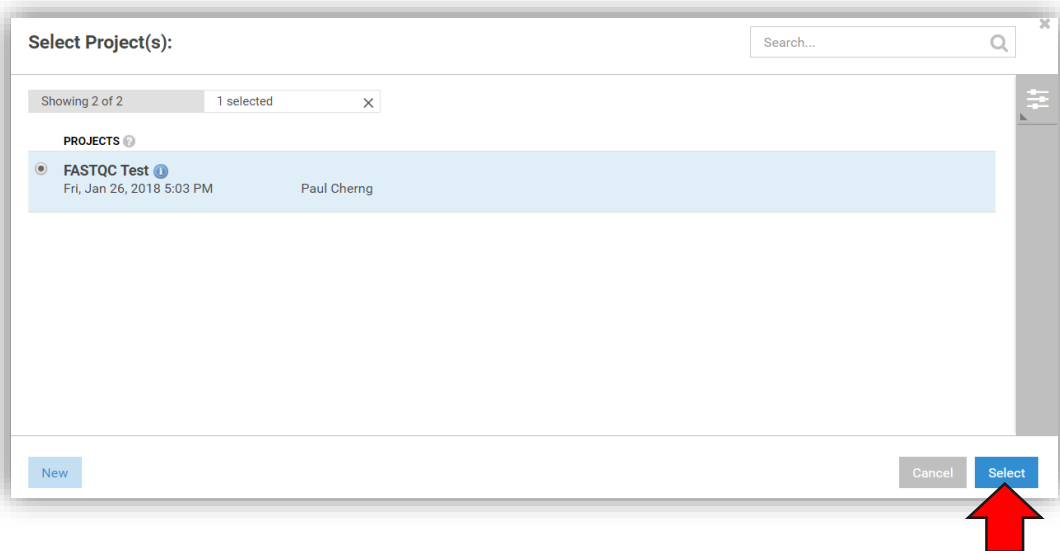
3<sup>rd</sup>





# Test the app

- Select the new “FASTQC Test” Project



# Test the app

- In the Input Form preview, click on “Select Biosamples(s)”

FASTQC Demo - Paul Cherng v1.0.0  
Illumina, Inc.

**Configuration**

**Analysis Name**  
FASTQC Demo - Paul Cherng 01/26/2018 4:54:30

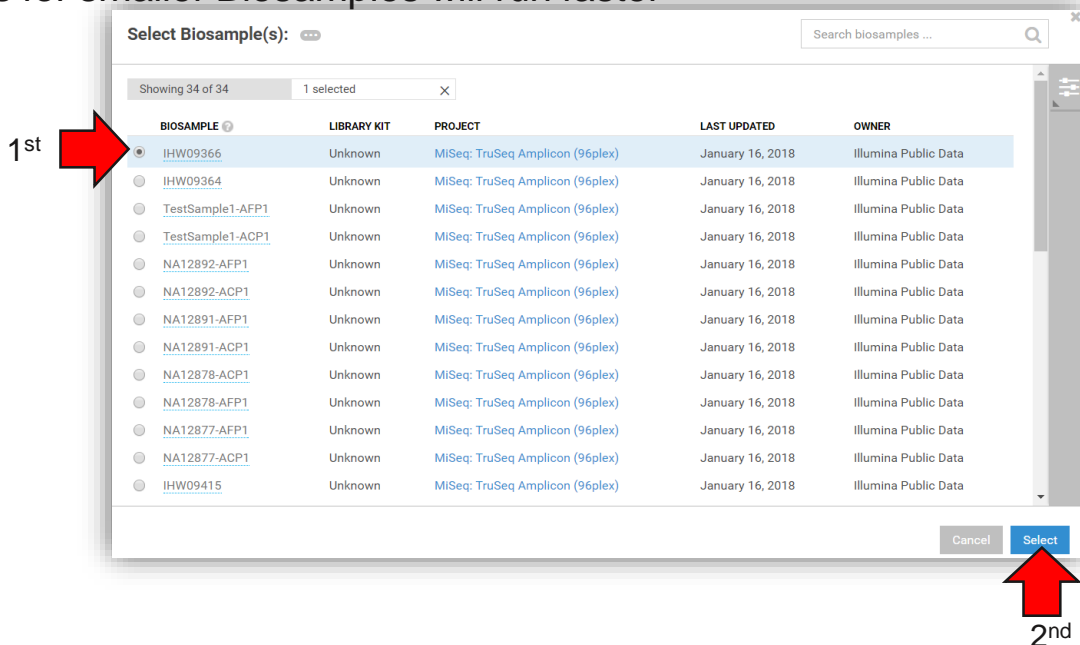
**Save Results To**  
Select Project(s):  
FASTQC Test

**Biosample**  
Select Biosample(s):

Launch Application

# Test the app

- **Select any Biosample**
  - Analysis for smaller Biosamples will run faster



# Test the app

- Click “Launch Application” to start the app
  - Make sure pop-up blockers are turned off

FASTQC Demo - Paul Cherng v1.0.0  
Illumina, Inc.

Configuration

Analysis Name

FASTQC Demo - Paul Cherng 01/26/2018 4:54:30

Save Results To

Select Project(s):

FASTQC Test

Biosample

Select Biosample(s):

IHW09366: Unknown

Launch Application

For Research Use Only. Not for use in diagnostic procedures.

# Test the app

- If successful, the Status should be “Complete”
- Click on “Output Files”



**FASTQC Test: FASTQC Demo - Paul Cherng 01/26/2018 5:19:44**

**ANALYSIS INFO**

**INPUTS**

**OUTPUT FILES**

**ANALYSIS REPORTS**

IHW09366\_S56\_L001\_R2\_001.FA  
STQ.GZ

IHW09366\_S56\_L001\_R1\_001.FA  
STQ.GZ

**Analysis Info**

Name	FASTQC Demo - Paul Cherng 01/26/2018 5:19:44
Application	FASTQC Demo - Paul Cherng   Version: 1.0.0
Date Started	Fri, Jan 26, 2018 5:20 PM
Date Completed	Fri, Jan 26, 2018 5:23 PM
Duration	3 minutes 11 seconds
Compute Charge	1.00 iCredits
Session Type	Single Node
Size	2.32 MB
Status	<a href="#">Complete</a>
Delivery	--

**Logs** (last checked 1:24:17 AM UTC)

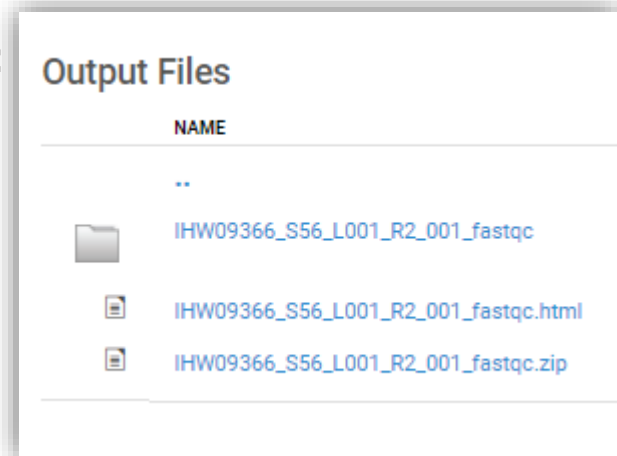
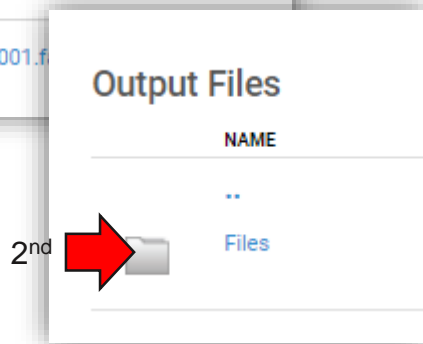
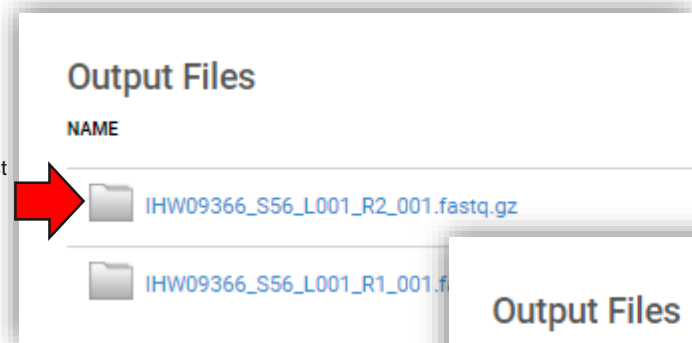
[Log Files](#)

```
2018-01-27 01:22:08.908 [INFO] - bsfs: /data/242244/data/input
2018-01-27 01:22:08.908 [INFO] - bsfs: -f
2018-01-27 01:22:08.908 [INFO] - bsfs: -f
```

For Research Use Only. Not for use in diagnostic procedures.

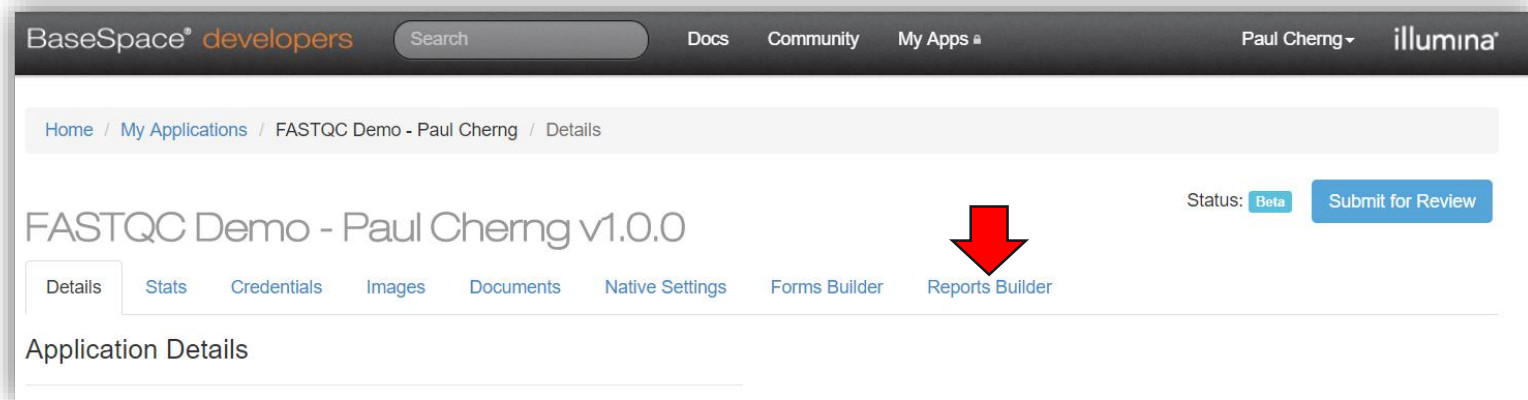
# Test the app

- Click on an AppResult, then the “Files” folder
- If successful, you should see outputs similar to this:



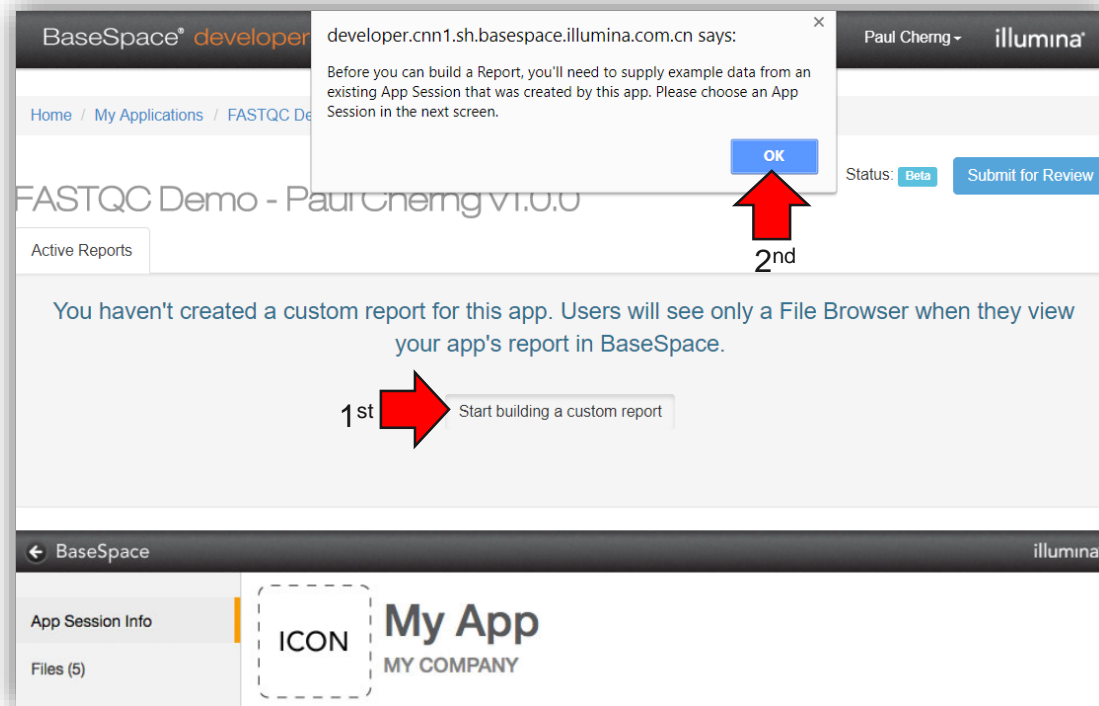
# Create a custom report template

- Return to the Developer Portal and click on the “Reports Builder” tab



# Create a custom report template

- Click “Start building a custom report”
- Click “OK” in the alert box

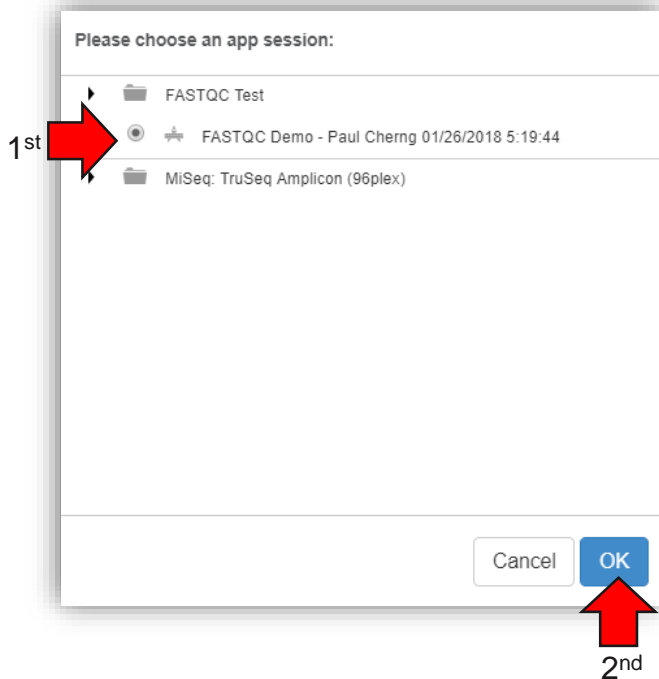


For Research Use Only. Not for use in diagnostic procedures.



# Create a custom report template

- Select the successfully completed test AppSession
- Click “OK”



# Create a custom report template

- Check the checkbox to “Use a custom report”
- Click on the link for the “Example” report template

BaseSpace<sup>®</sup> developers Docs Community My Apps Paul Cherng

[Home](#) / [My Applications](#) / [FASTQC Demo - Paul Cherng](#) / Reports Builder

FASTQC Demo - Paul Cherng v1.0.0 Status: [Submit for Review](#)

Revisions

1st ☒ Use a custom report (below). If this option is unchecked, users will only be able to view results of your app session by browsing through the file browser.

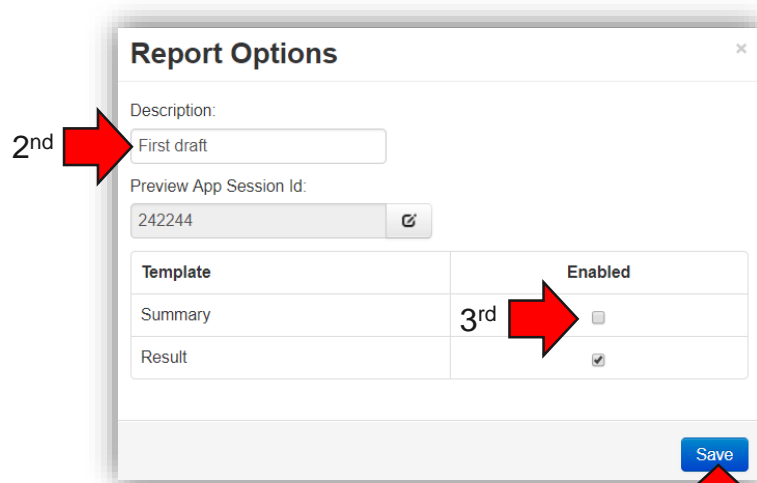
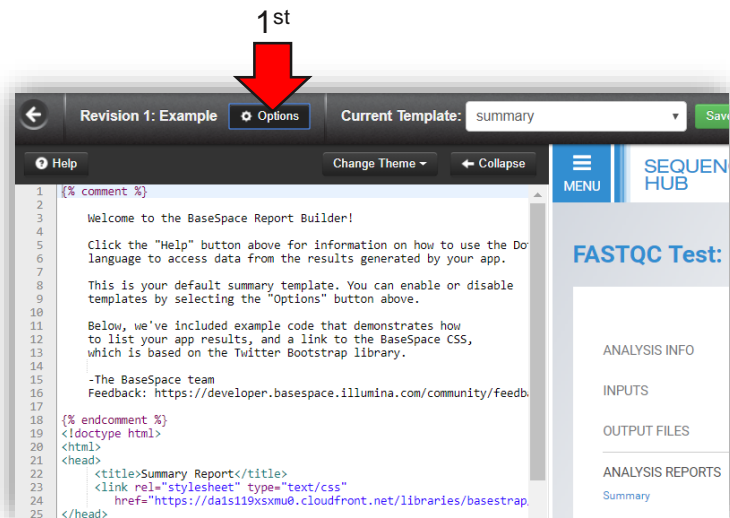
Version	Description	Publish Status	DateModified
1	<a href="#">Example</a>	Inactive	in a few seconds

2nd

For Research Use Only. Not for use in diagnostic procedures.

# Create a custom report template

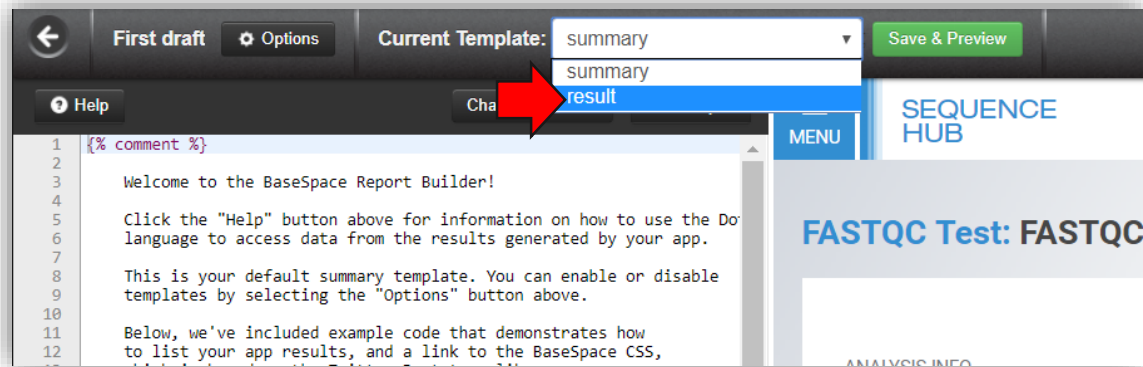
- Click on the “Options” button
- Rename the Description to “First draft”
- Uncheck the “Summary” checkbox
- Click “Save”



For Research Use Only. Not for use in diagnostic procedures.

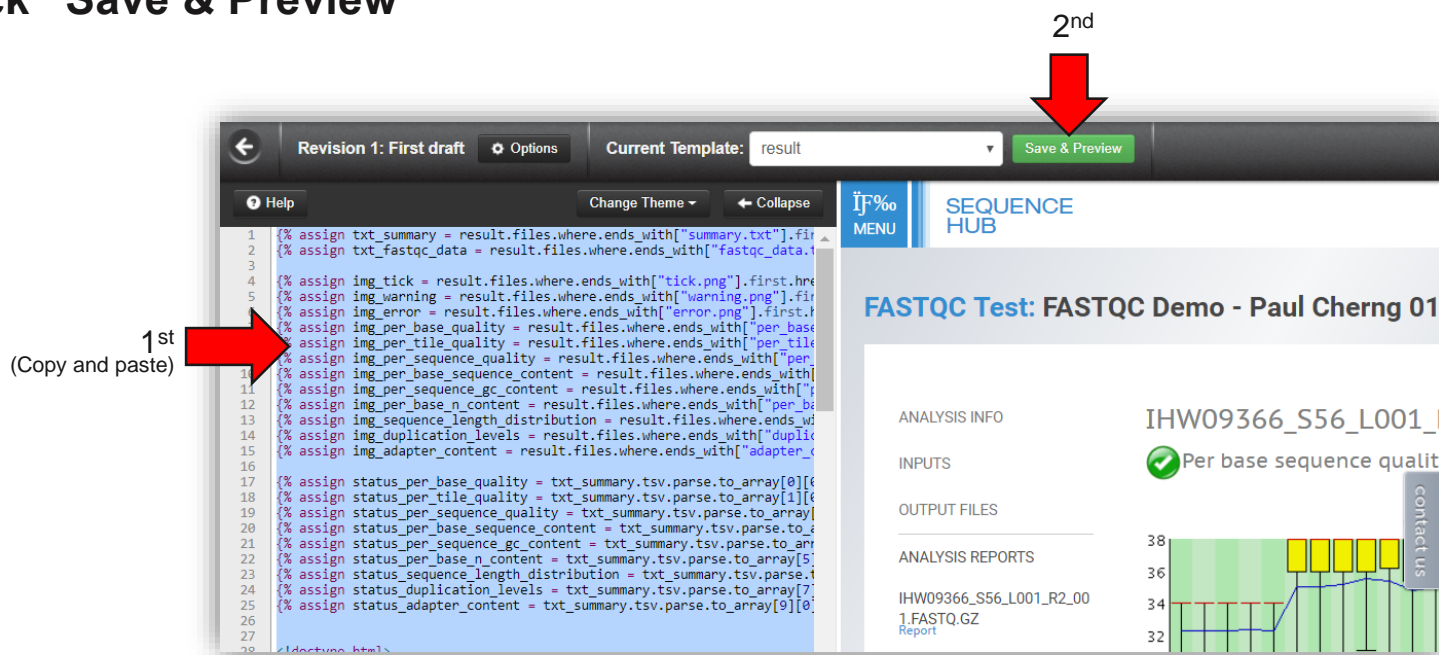
# Create a custom report template

- Change the Current Template to “result”



# Create a custom report template

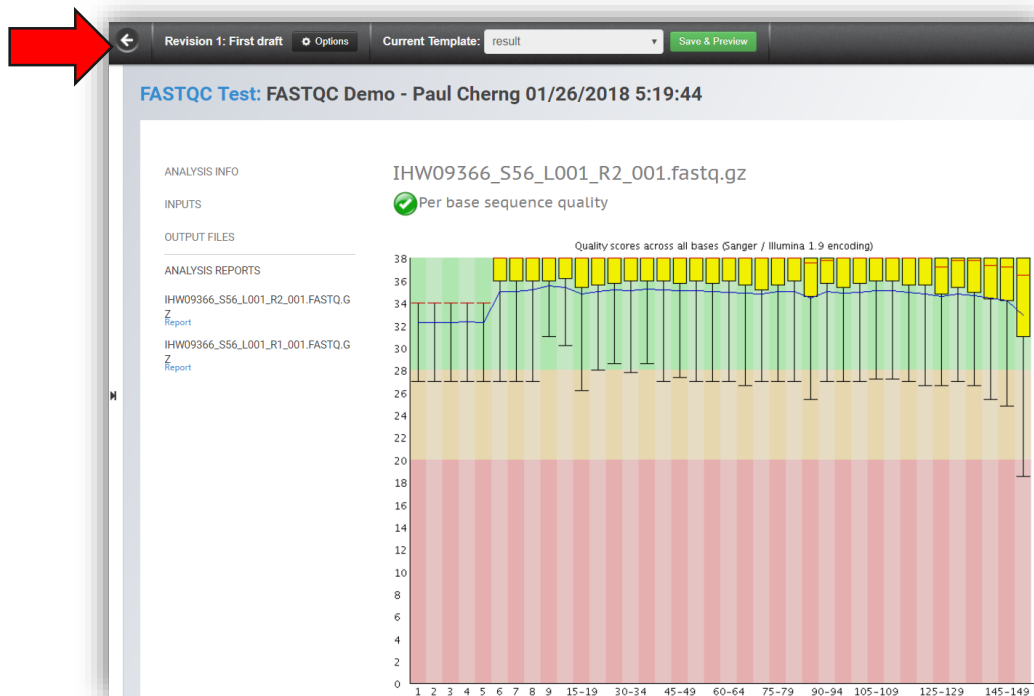
- Copy and paste “result.html” content from “05\_Report\_Builder” folder in flash drive
- Click “Save & Preview”



For Research Use Only. Not for use in diagnostic procedures.

# Activate custom report template

- If successful, report should look similar to below
- Return to the Revisions page by clicking the “←” button



# Activate custom report template

1. Click the “Use a custom report” checkbox again
2. Select the “First draft” report template
3. Click the “Activate” button

BaseSpace<sup>®</sup> developers Search Docs Community My Apps Paul Cherng illuminat<sup>®</sup>

Home / My Applications / FASTQC Demo - Paul Cherng / Reports Builder

FASTQC Demo - Paul Cherng v1.0.0 Status: Beta Submit for Review

Revisions

1<sup>st</sup> ☒ Use a custom report (below). If this option is unchecked, users will only be able to view results of your app session by browsing through the file browser.

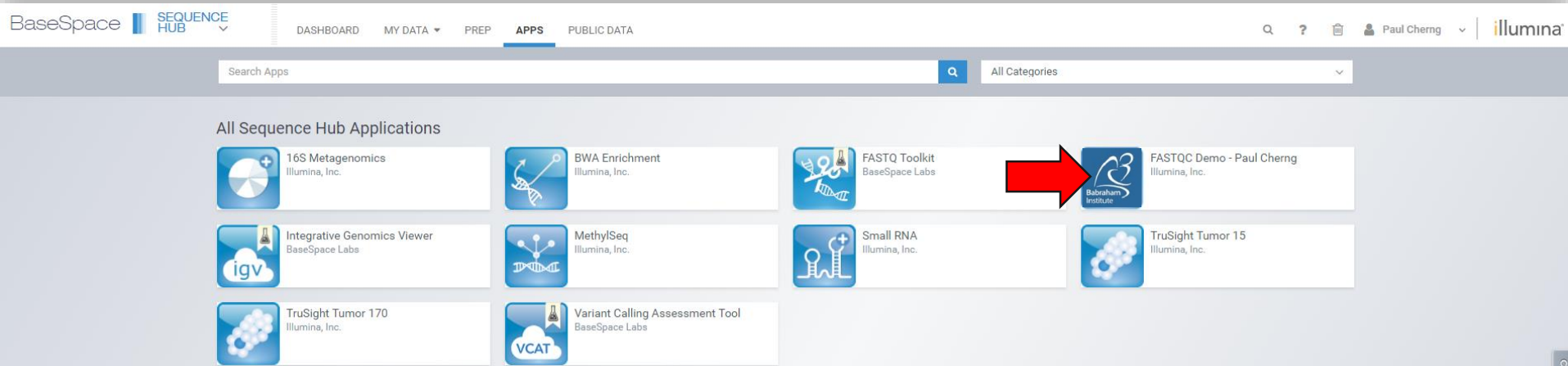
3<sup>rd</sup>

Version	Description	Publish Status	DateModified
1	First draft	Inactive	10 minutes ago

For Research Use Only. Not for use in diagnostic procedures.

# Congratulations!

- You have now successfully created your first BaseSpace Sequence Hub Native App!
- Your app should be visible (only to yourself) in the BaseSpace Sequence Hub App Store
  - Additional users and workgroups can be added in the Developer Portal





## Closure ceremony



# Hand out certificates

# Appendix



# Documentation

- **App Development**

- <https://developer.cnn1.sh.basespace.illumina.com.cn/docs>

- **Input Form**

- <https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/apptools/formbuilder-overview>

- **Docker Images**

- <https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/native-apps/manage-docker-image>
- <https://docs.docker.com/>

- **Report Builder**

- <https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/apptools/reportbuilder-overview>

# Create Docker Image

- Copy “04\_Docker\_Image” folder from flash drive to Linux server with Docker installed. This folder contains all the files needed to build the Docker Image.
- Login to Illumina Docker registry with this command:
  - `docker login docker.cnn1.sh.basespace.illumina.com.cn`
- Enter BaseSpace Sequence Hub username and password when prompted

```
-bash-4.2$ docker login docker.cnn1.sh.basespace.illumina.com.cn
Username: pcherng@illumina.com
Password:
Login Succeeded
```

# Create Docker Image

- **Change directory to the 04\_Docker\_Image folder**
  - Example: `cd ./04_Docker_Image`
- **Build the Docker image from the included Dockerfile with this command:**
  - `docker build -t <docker image name> .`
  - **<docker image name>** should be replaced with whatever you entered for containerId in the Input Form callbacks.js script
  - Example:
    - `docker build -t docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app .`

```
pcherng@ussd-prd-dock01:~$ cd 04_Docker_Image/  
pcherng@ussd-prd-dock01:~/04_Docker_Image$ sudo docker build -t docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app .  
Sending build context to Docker daemon 20.48kB  
Step 1/6 : FROM centos:centos7  
--> d123f4e55e12
```

```
Step 6/6 : COPY fastqc_app_with_kmer.py /fastqc_app_with_kmer.py  
--> Using cache  
--> 59e335733c1c  
Successfully built 59e335733c1c  
Successfully tagged docker.cnn1.sh.basespace.illumina.com.cn/pcherng/fastqc-app:latest
```

agnostic procedures.



# If you have VM installed ...

- **Instructions to load BaseSpace Native App VM:**
    - <https://developer.cnn1.sh.basespace.illumina.com.cn/docs/content/documentation/native-apps/setup-dev-environment>
- 
- 1) **Install Virtual Box on your Windows / Mac**
  - 2) **Load BaseSpaceNativeApp.ova into Virtual Box**
  - 3) **Login with basespace/basespace**

# Instructions for Mac or Linux

## Install Virtual Box and Import the Native Apps Virtual Machine on Mac or Linux

1. Install VirtualBox
  - a. Select the VirtualBox installer for Mac
  - b. Click through the dialogs and install VirtualBox
  - c. Restart your computer
  - d. Find "Oracle VM VirtualBox" in your applications and open the program
2. In the VirtualBox window, select **File -> Import Appliance...**
3. Click **Open Appliance...**
4. Select the **BaseSpace Native App VM.ova**
5. Click **Next**
6. Ensure that the **Reinitialize the MAC address of all selected network cards** option is selected
7. Click **Import**, this will start importing the VM and the process may take several minutes
8. Verify that the Virtual Machine was successfully imported by selecting the new VM from the list in Virtual Box and clicking **Start**. A new window will open and begin loading your VM, if you are prompted to enter a password you have completed setting up your local dev environment.

## SSH into your local Virtual Machine

1. On your local machine, open a terminal by launching the **Terminal** application
2. In your terminal's command line, please enter the following: **ssh basespace@localhost -p2222** and hit **Enter**
3. You will be prompted for a password, the password is **basespace**



## Install Virtual Box and Import the Native Apps Virtual Machine on Windows

1. Install VirtualBox
  - a. Select the appropriate VirtualBox installer for your OS
  - b. Click through the dialogs and install VirtualBox
  - c. Restart your computer
  - d. Find "Oracle VM VirtualBox" in your applications and open the program
2. In the VirtualBox window, select **File -> Import Appliance...**
3. Click **Open Appliance...**
4. Select the **BaseSpace Native App VM.ova**
5. Click **Next**
6. Ensure that the **Reinitialize the MAC address of all selected network cards** option is selected
7. Click **Import**, this will start importing the VM and the process may take several minutes
8. Verify that the Virtual Machine was successfully imported by selecting the new VM from the list in Virtual Box and clicking **Start**. A new window will open and begin loading your VM, if you are prompted to enter a password you have completed setting up your local dev environment.

## SSH into your local Virtual Machine

1. Download and install an SSH client if you do not already have one. Some examples of popular SSH clients: [PuTTY](#), [OpenSSH](#), and [WinSCP](#)
2. Open your SSH client, the following example will be using PuTTY
3. Fill in the following information:  
Host Name (or IP Address): **basespace@localhost** Port: 2222
4. Click **Open**
5. A terminal will open and you will be prompted for a password, the password is **basespace**

# What is spacedock?

- spacedock can also be run locally using the development VM for testing and debugging purposes

Tools

Billing spec console 0

This section is only visible to the developer ↑

Sample command-line to start local agent:

```
sudo spacedock -a c26b6752c8b7498983cba8690068aee5 -m https://mission.cnn1.sh.basespace.illumina.com.cn
```

Send to Local Agent

Create AppSession

Simulate Launch

What is this?