

# iCompute.us Web Portal User Guide

Welcome to the iCompute.us web portal, your gateway to utilizing **Basil, which is a powerful tool for semi-automatic containerization, deployment, and execution of applications and workflows on various cloud computing and supercomputing platforms.** This user guide will walk you through the process of using Basil via the iCompute.us web portal.

## Table of Contents

1. Introduction to Basil
2. Getting Started
3. Creating Docker Images
4. Testing and Deploying Images
5. Managing Images
6. Interacting with Basil via Command Line
7. Troubleshooting
8. FAQs

## 1. Introduction to Basil

Basil is a tool that simplifies the process of creating optimized and secure Docker images without requiring in-depth knowledge of Docker image creation. It accepts recipes for building code, generates Dockerfiles on the basis of the recipes, builds Docker images, scans for vulnerabilities, signs the images, and supports pushing images directly to DockerHub if needed. The Docker images generated using Basil can then be run on diverse hardware platforms that support Docker or Singularity. Basil also provides a mechanism for adding an appropriate license file to the Docker images before they are shared publicly. It generates README files as well which contain the instructions on using the Docker images generated by Basil.

The homepage of the Basil website has the information regarding the current functionalities available as well as new updates listed under the “Project News” section.

As of August 23, 2023, the first iteration of Basil development is complete and we are in the beta testing phase of our Minimum Viable Product (MVP) version. In the second iteration, we will add support for providing the software build recipes in additional formats such as: Makefiles and CMakefiles. Currently, Basil can accept the recipes for building the code in the form of bash scripts, commands pasted in the text box provided on its website, or by using predefined templates (provided as YML files). We will also provide the support for generating Singularity image files directly through Basil (currently Basil can generate Docker images which can then be pulled as Singularity images).

In the third iteration of the Basil development, we will also provide the feature for running the generated Docker/Singularity images on remote supercomputing and cloud computing platforms from the convenience of the web browser.

## 2. Getting Started

Accessing iCompute.us: Visit [icompute.us](https://icompute.us) and login using your credentials. There are two ways to login, (1) you can either create a new account on the Basil web portal that is linked with your email address, or (2) you can login using CILogon and choose your existing accounts on GitHub, Google/Gmail, or your institutional account accounts to proceed.

## 3. Creating Docker Images

Once logged in, navigate to either the "Terminal" tab or the "iSpec" tab.

In the "Terminal" tab, you can interactively provide specifications for building a Docker image by (1) uploading the necessary script and software files as a project folder or directory, (2) navigating to that directory and then, (3) typing *basil -b*. Complete the required input sections with the correct information regarding your software and a Dockerfile will be generated and shared with you right in your project directory. You can also choose to push the image to a Docker Hub account. If you choose to not push your image to Docker Hub then a link to download the image will be sent to you over email. There is a video demonstration of this process available under the "Help" tab (please select the "Video Demos" option).

In the "iSpec" tab, you can specify the recipe for image creation using scripts, commands, or predefined text templates. Upload any relevant files (source code and input files) needed for building the Dockerfile and follow along the process that involves filling out necessary input fields to build your Dockerfile. In the end you can choose to receive your Dockerfile in one of two ways:

- a. Push the generated Docker image to a Docker Hub account by entering the username and access token.
- b. Receive an email with the link to download the generated Docker image.

## 4. Testing and Deploying Images

Generate Dockerfiles and images: After specifying the recipe, click the appropriate button to generate the Dockerfile. In the iSpec tab, you will be shown a completed YAML file based on your input, which is used by Basil to make the Dockerfile and generate the Docker image in the background. The customer will be sent an email regardless of the action taken and the email will contain information and steps on what to do next relevant to your choices.

Build Images again using the Dockerfiles generated by Basil: Basil will automatically build the image using the Dockerfile that it generates. The image will either be sent to a Docker Hub account or shared with you via a link to download the TAR file that contains the image. For more information on how to run the image using both methods, a video demonstration is available under the “Help” tab by selecting “Video Demos”.

Deployment: Images can be deployed or run on users’ systems or in the cloud or on supercomputing platforms based on customer needs.

## 5. Managing Images

View Images: You can view your generated images in either the Docker Hub account it was pushed to, or you can check your emails for an email by ([icomputeus@gmail.com](mailto:icomputeus@gmail.com)) that will include some attachments as well as a link to download the TAR file, that is the image.

Scan for Vulnerabilities: Basil automatically scans generated images for vulnerabilities, you can review the scan results and address any issues. The vulnerability analysis run by SNYK is attached as a text file named ‘vulnerability\_analysis.log’ in the email sent to you after generation.

Sign and Release: If desired, you can sign images and release them in public registries with appropriate licenses.

Add licenses: If desired, you can add a license to your Docker image.

## 6. Running created Docker image in user environment

If you wish to build a Docker image on the system of your choice using the Dockerfile generated by Basil, then please make sure that you have Docker or Singularity installed on the system.

Please review the steps below to install Docker. Docker is required even if you would just like to run the Docker image, so it is a good idea to install it beforehand.

### Steps for installing Docker on Windows:

1. Go to the Docker website: <https://www.docker.com/get-started>
2. Download the Docker Desktop Installer.exe file
3. Run the .exe file to run and finish the installation by following the on-screen instructions

### Steps for installing Docker on macOS:

1. Open a terminal window
2. Run the following command to download the installer for Docker: `curl -fsSL https://get.docker.com -o get-docker.sh`
3. Now run the following command to install Docker: `sudo sh get-docker.sh`

## Steps for installing Docker on Linux:

1. Open a terminal window
2. Run the following command: `sudo apt-add-repository https://download.docker.com/linux/ubuntu/repo/deb`
3. Next, run this command: `sudo apt update`
4. Now, run this command: `sudo apt install docker-ce`
5. Finally, run this command to check if Docker has been installed correctly: `docker version`

## #1 INSTRUCTIONS FOR RUNNING DOCKER IMAGES GENERATED BY BASIL ON THE SYSTEM OF YOUR CHOICE

### Case 1: The Docker image is available in your or someone else's Docker Hub account

- Step 1: Go to the Docker Hub account from which you would like to pull a Docker image, locate the image of your choice, and copy the Docker Pull Command provided on the page. For example: `docker pull basilproject/alpine`.
- Step 2: Paste the command in a terminal session and hit enter, it should pull the image from Docker Hub to your system.
- Step 3: Run the Docker image using the docker run command as follows: `'docker run replace-this-with-your-image-name'` and it should run your image and perform the task that it was programmed to do.

### Case 2: Your or someone else's Docker image that may be available in Basil project's Docker Hub account

- Step 1: Go to <https://hub.docker.com/> and type 'basilproject' in the search bar located in the top left part of the website. You do not need to have a Docker Hub account to perform this action. Step 2: Locate the image that was pushed here, if you selected to push your image, the image name will have your username in it to make it easier to identify the image.
- Step 3: Copy the Docker Pull Command provided on the webpage and paste it into your terminal session and hit enter.
- Step 4: Once the pull has completed, run `'docker run [image name]'` command and it should run your image.

### Case 3: Did not push the image to Docker Hub

- Step 1: Check your email, you should have an email from 'icomputeus@gmail.com'. It should have a link to download the file as well as instructions on how to run it.
- Step 2: Click the link to download the file, once the file has downloaded, open terminal and navigate to the folder where the .tar.gz file is located.
- Step 3: Run this command, `docker load < [image name.tar.gz]` and wait for the image to be loaded.
- Step 4: Confirm the image is available on your system by running `'docker image ls'`.
- Step 5: Run the command, `'docker run [image name]'`, and you are done!

## #2 INSTRUCTIONS FOR BUILDING THE DOCKER IMAGE FROM THE BASIL GENERATED DOCKERFILE

Note: You will need "sudo"/"root"/"admin" level permissions on the system on which you need to build a Docker image.

- Step 1: If you used the iSpec form on the Basil website to generate the Docker image, then you can download the Dockerfile from the link provided in the email notification you received from the Basil project. If you ran the "Basil" tool on the Basil website through the terminal tab, then you can click the "Refresh List" button on the terminal page, select the Dockerfile that you wish to download from the drop down menu, and click on the "Download" button.
- Step 2: To build an image based on the instructions on the Dockerfile, run the following command `'docker build .'` from the folder containing the Dockerfile.
- Step 3: Once the docker image is built successfully, type `'docker run [image name]'` to run the image.

## #3 INSTRUCTIONS FOR RUNNING THE IMAGE ON HPC SYSTEMS USING SINGULARITY

- Step 1: If you are working on an HPC system that is provisioned with the module system, you can run the "module avail" command and check if Singularity is already installed. If Singularity is available as a module, you can load it by using the `"module load [singularity-module-name]"` command.
- Step 2: Use the following command to pull an image from Docker Hub and convert it into a Singularity Image Format (SIF) file. `'singularity pull [image name].sif docker://[docker image name]:[tag]'` # remove [ ] in actual use
- Step 3: Run the Singularity image by using the command as follows `'singularity run [image name].sif'`

PLEASE NOTE: If you already have the image, you may not need to rebuild it unless there are any changes that were made to the previous image. You can confirm if the image is available on your system by running the `'docker image ls'` command in the terminal.

## 6. Interacting with Basil via Command Line

Command-Line Mode: You can interact with Basil via the command line using the "Terminal" tab. Execute commands to specify image specifications, generate files, build images, and more. To start this process, simply head to your project directory and type the command `'basil -b'`.

## 7. Troubleshooting

If you encounter issues during any step, please refer to the following resources:

- Documentation: Check the documentation section of iCompute.us for relevant instructions.
- Support: Reach out to the iCompute.us support team for assistance by posting a message on the Basil website's (icompute.us) messageboard.

- Video Demos: Listed under the “Help” tab, this section has a set of video demonstrations that include various use cases and cover the functionalities offered by Basil.

## **8. FAQs**

For frequently asked questions, visit the "FAQ" section on the [iCompute.us](https://iCompute.us) website.

Congratulations! You've successfully learned the basics of using Basil through the [iCompute.us](https://iCompute.us) web portal. With Basil's powerful capabilities, you can simplify the containerization, deployment, and/or execution of your applications and workflows on various computing platforms.