

3 hour tutorial

The quickstart needs to be made to work with 2.7

source data:

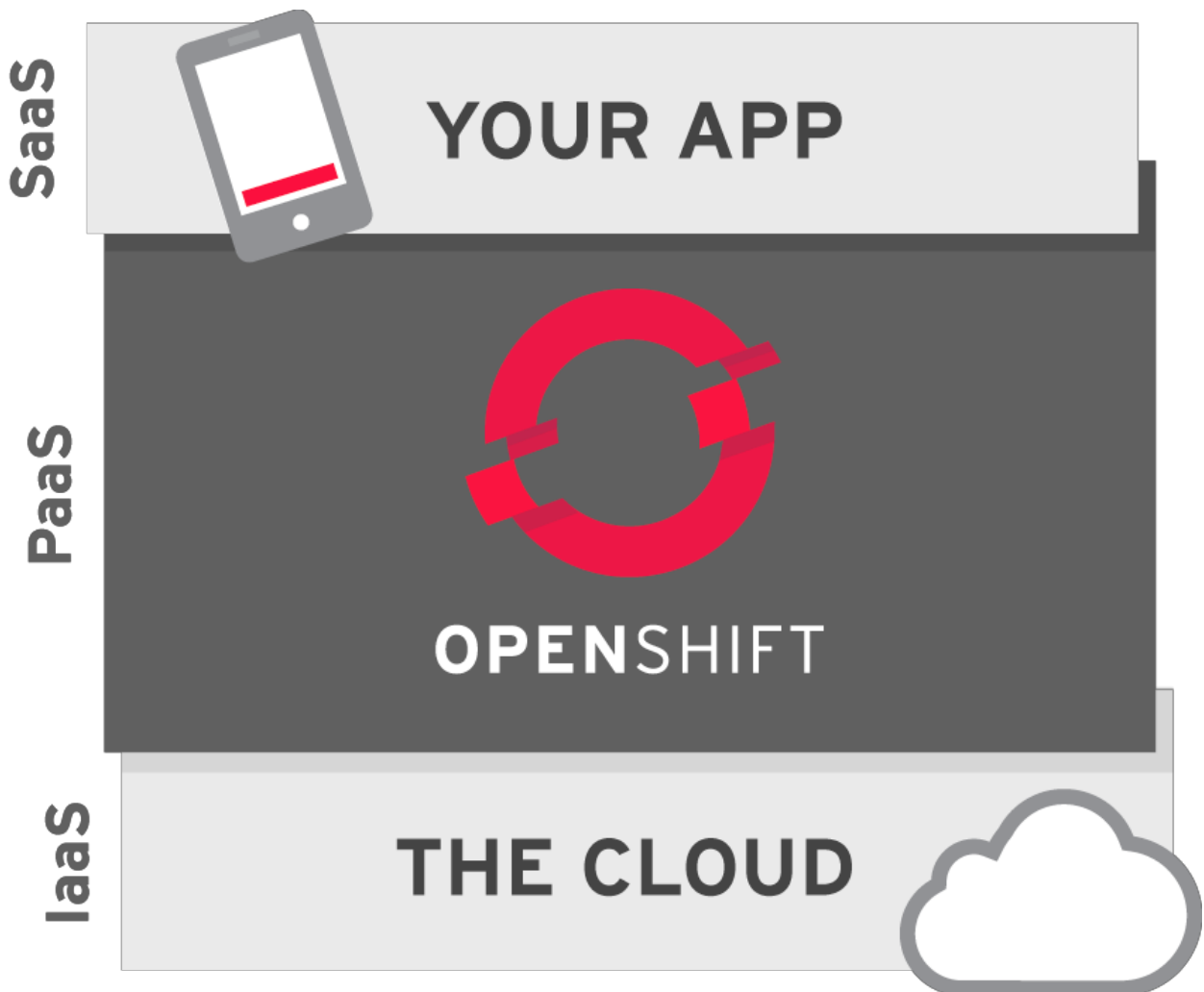
http://gnis.usgs.gov/domestic/download_data.htm

- Concise Features – Large features that should be labeled on maps with a scale of 1:250,000. Subset of National file above.

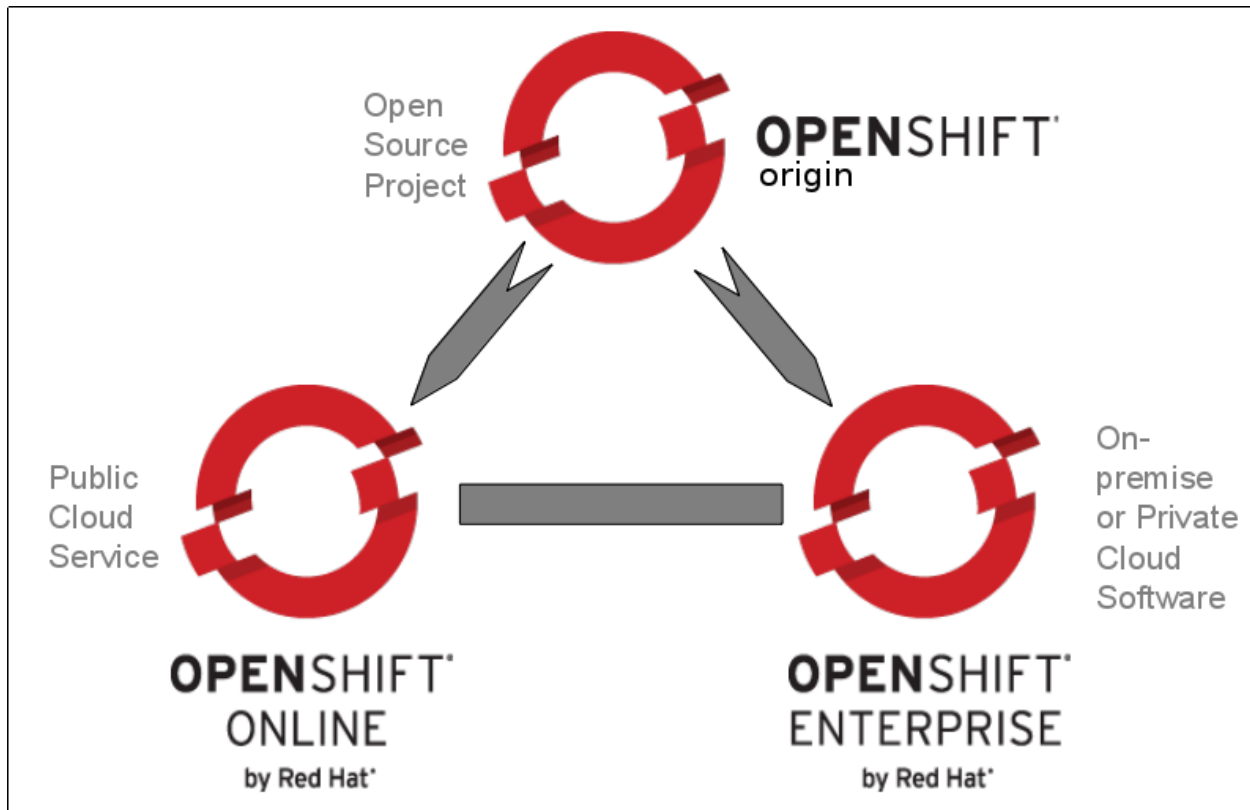
(last updated October 2, 2009)

1. Introduction of us, quick check of the who is in the room, and ground rules
2. Make sure everyone has an OpenShift account and they are ready to go

PaaS



OpenShift

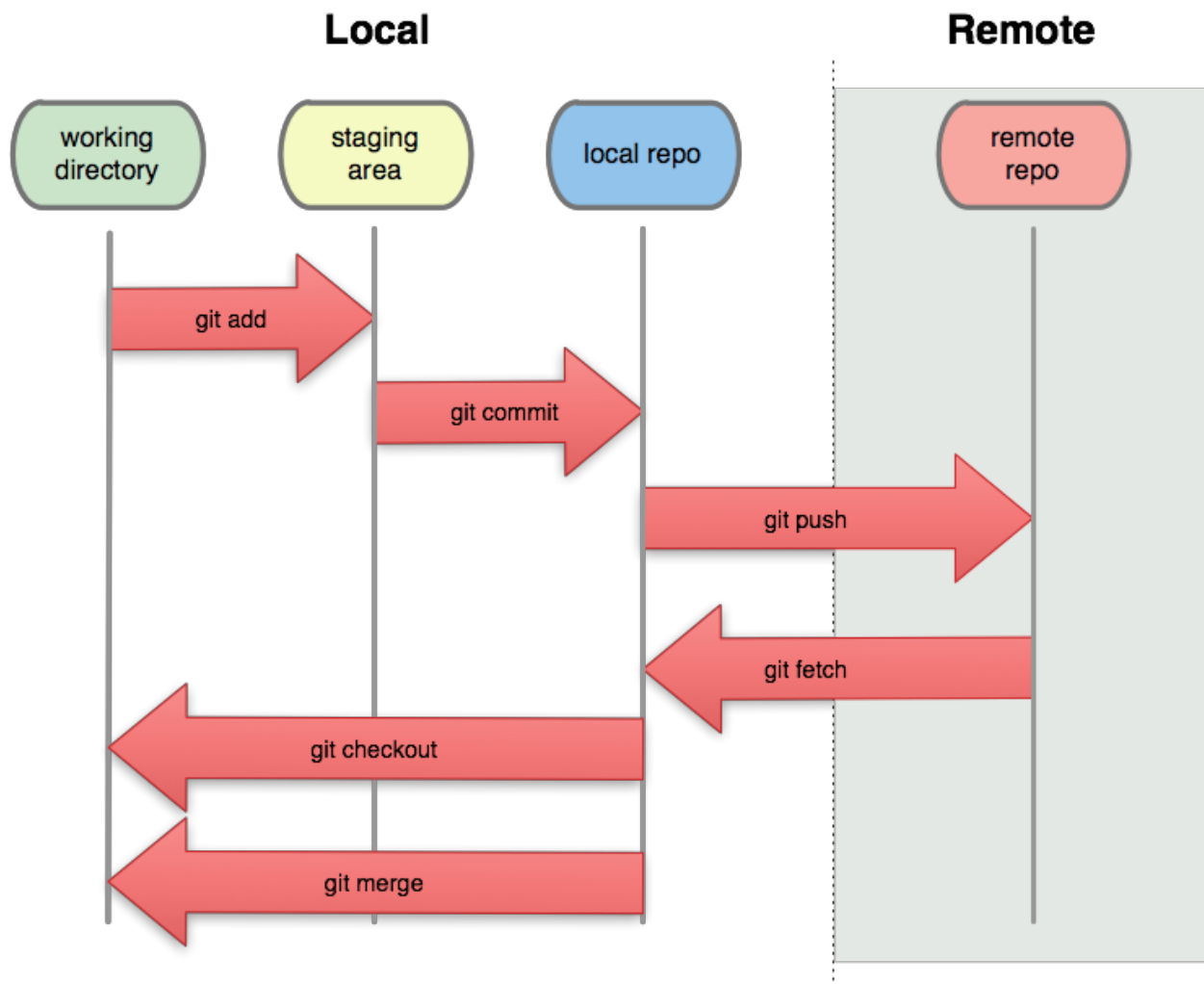


RHC Command Line Tools

We are going to use the command line tools throughout the day today. These tools are a Ruby command line program that talks to the REST interface on OpenShift. Once you install them you access them using the 'rhc' command. If you ever need help there is a lot of help on the tools, just do `rhc --help`. You can also use this on subcommands, so for example you can do `rhc app --help` to get help on commands pertaining to managing applications. If you clone your application git repository using the rhc tools then you usually don't have to enter the app name with your commands. You just need to execute the commands in the git repo for the application; there is metadata in the git repo to tell the rhc tools enough that it can figure out the application.

GIT

Love this simple guide <http://rogerdudler.github.io/git-guide/>



3 Commands You need to know

git add = make a file part of your repository

git commit = commit your changes to your LOCAL repository

git push = push the changes from our local repository to a remote repository

SSH = Secure Shell into a linux machine

To transfer files from your local machine, you can also use scp or sftp. If you are on windows (or if you just prefer a graphical tool) you can use FileZilla. Here are instructions on how to set it up with OpenShift -

<https://www.openshift.com/blogs/using-filezilla-and-sftp-on-windows-with-openshift>

There is also an rhc command for scp - *rhc scp*

3. Create a Python applications (2.7)

rhc create-app pythonwebmap python-2.7 -s -g medium

- a. To see a list of all cartridges do *rhc cartridge list*
- b. To see the community cartridges go to <http://origin.ly>
- c. You could also use large.
- d. if you leave off the -g medium then you will get a small gear by default
- e. Modify
 - i. Just go into the repo, take your favorite editor, and then edit the wsgi.py.
 - ii. Search for h1 tag, change the inside element.
 - iii. *git commit -am "whatever message you want"*
- f. *Git push*
- g. Go to the web page and see your page. You are now a Pousty Certified Cloud deployer

4. BREAK

5. Add Mongo to Python

rhc add-cartridge mongodb-2.4

- a. you can also use a -g on this command to change the gear size so it is NOT the same as the default one for the application
- b. SSH in and fire up the mongo prompt
 - i. *show dbs*
 - ii. *show collections*
 - iii. *make a collection on the fly*

6. The Github Repo - <https://github.com/thesteve0/pythonwebmap>

- a. Add it as an upstream
- b. Merge it in

cd pythonwebmap

git remote add upstream -m master <https://github.com/thesteve0/pythonwebmap.git>

git pull -s recursive -X theirs upstream master

git push

7. Intro to Mongo Spatial, leaflet, and flask

<http://talks.thesteve0.com/flaskmap>

8. You should now understand the basic pieces of the application

9. BREAK

10. Look at the code working together

- a. Action Hooks
 - i. OpenShift has a hidden directory in the git repo called .openshift. In there we can control a bunch of action.
 - ii. We used an action hook to load all the data into MongoDB and create the spatial index.
 - iii. Let's go look

- b. We also used file in the Marker to directory to stop Apache and MongoDB from stopping and starting on every git push
 - i. Let's go look again
- 11. How to Modify for your own purposes
 - a. Places that might need modification
 - i. Update the JSON file
 - ii. Change the import script
 - iii. If you used different attribute names you might have to change the Python
 - iv. You would probably have to make minor (or major) modifications to the JavaScript
 - b. For Fun and if there is time, I would love to take on two modifications
 - i. Reduce the fields we are returning so our JSON payload is smaller
 - ii. Cluster the points on the map so we can return more
 - iii. Make a heat map from the elevation values

12. Wrap up