Joshua Catoe

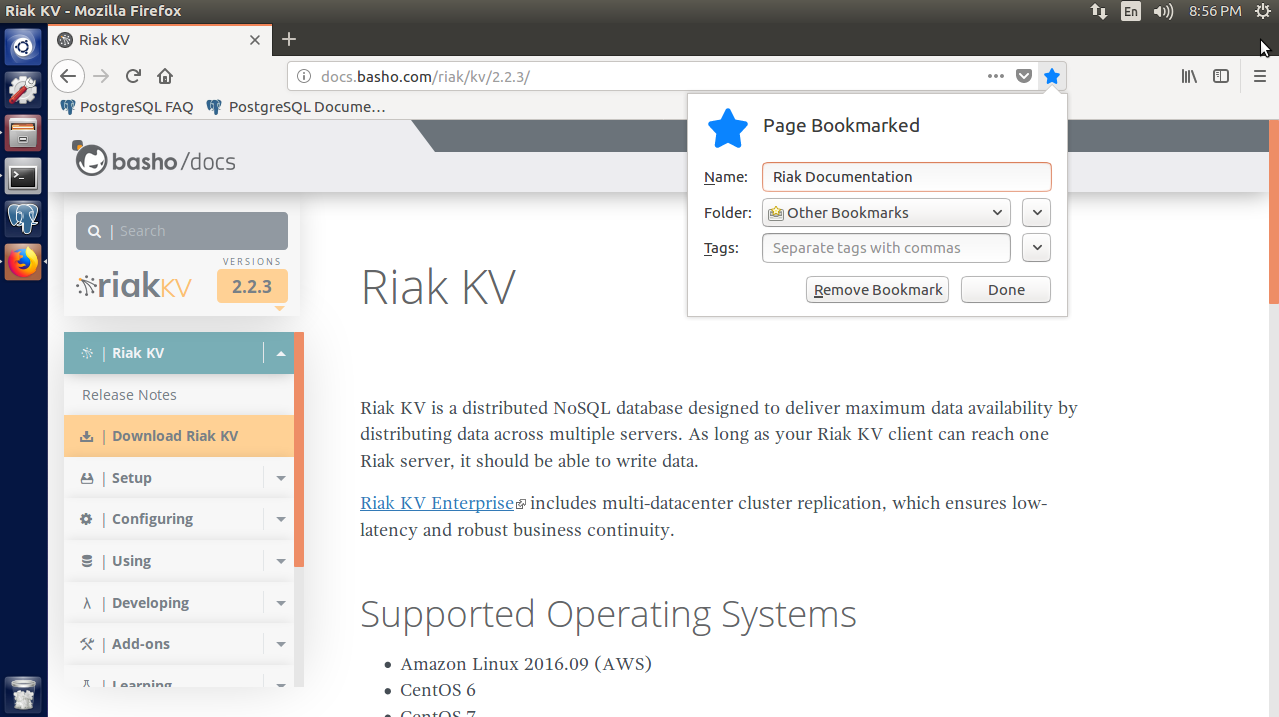
07/22/18

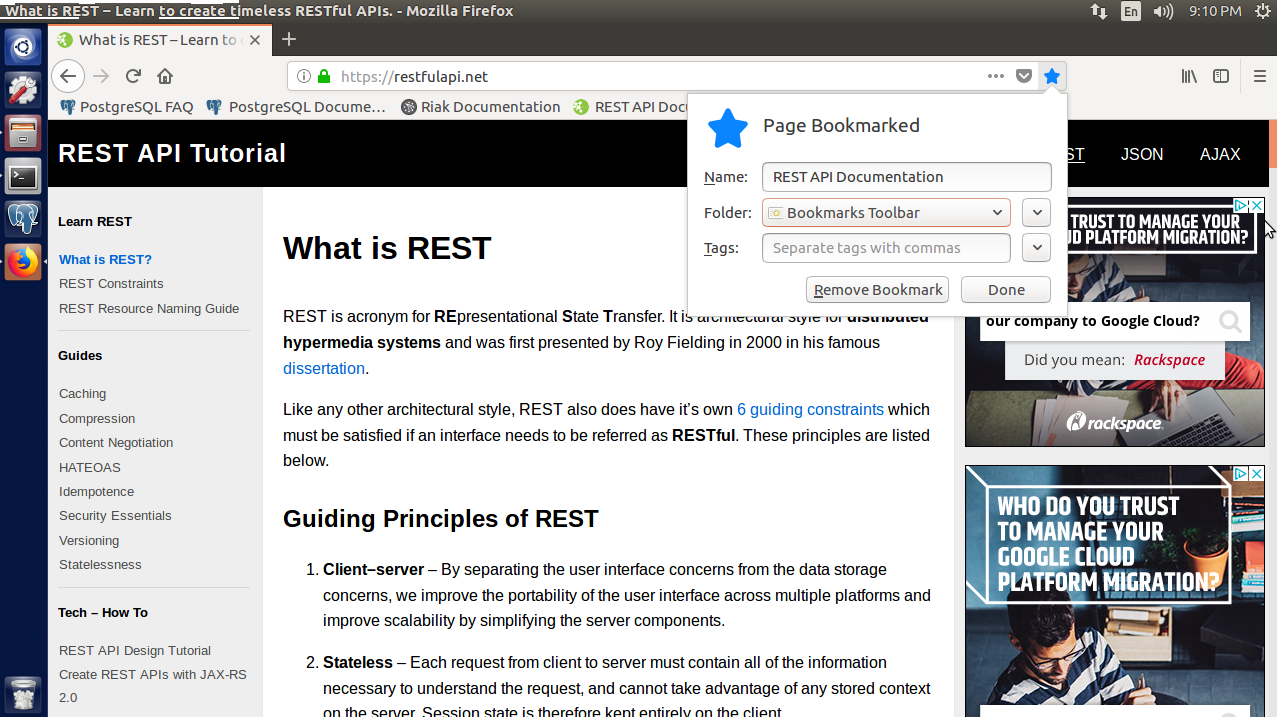
Riak

**Day 1**

Find:

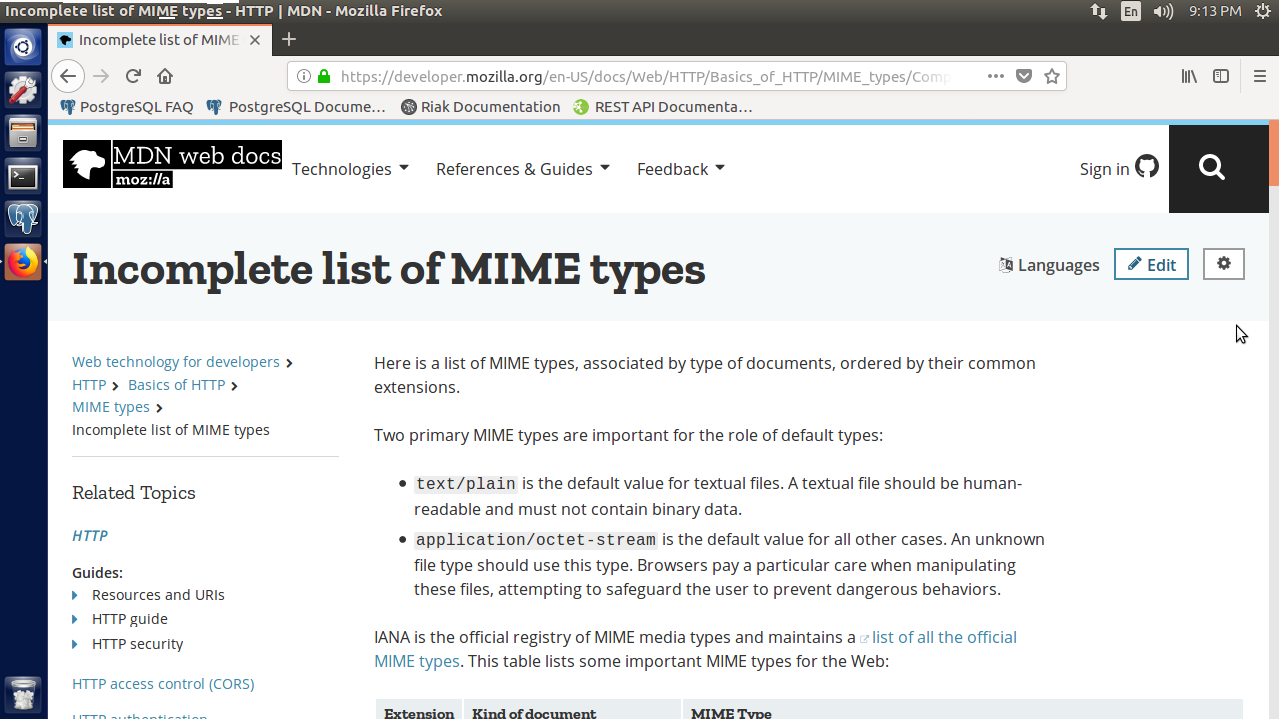
1. Bookmark the online Riak project documentation and discover the REST API documentation.





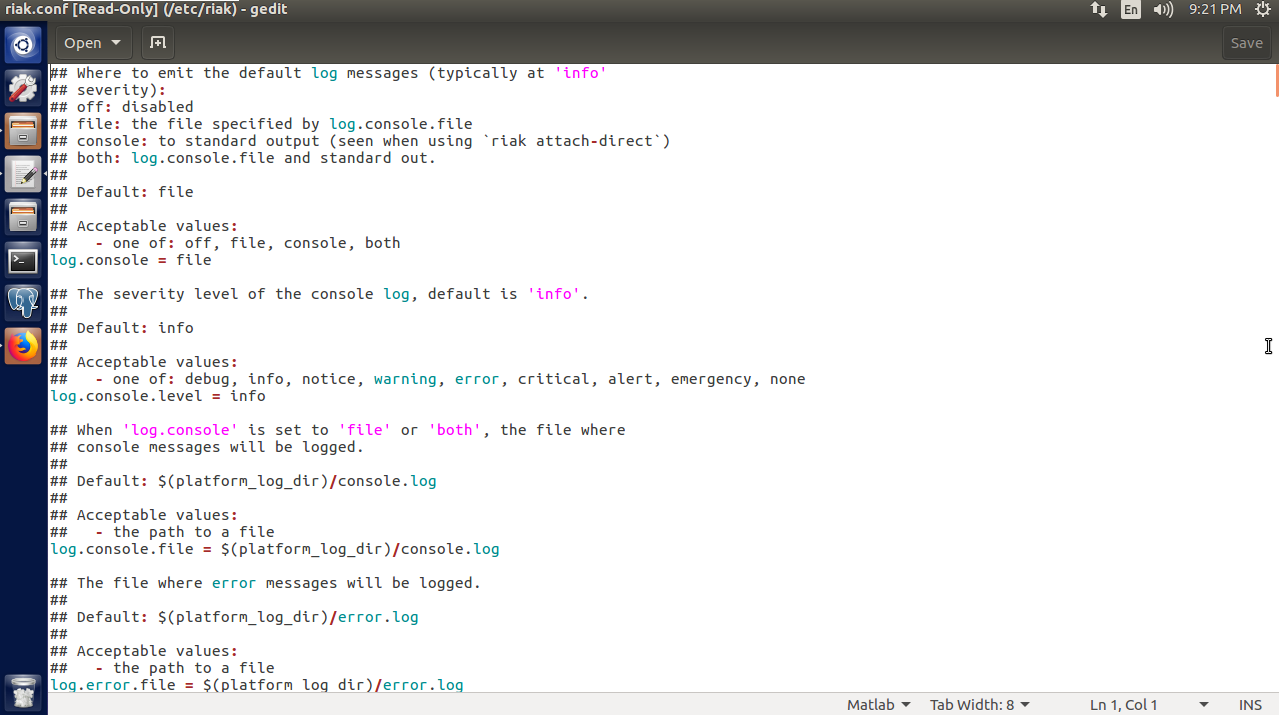
Documentation was easy to find with a Google search. I just used *Ctrl+D* to bookmark it. Riak, it turns out, has three different versions, but the one we’re using is called Riak KV.

2. Find a good list of browser-supported MIME types.



Through Google search, I found a list of supported MIME types from Mozilla (creators of Firefox).

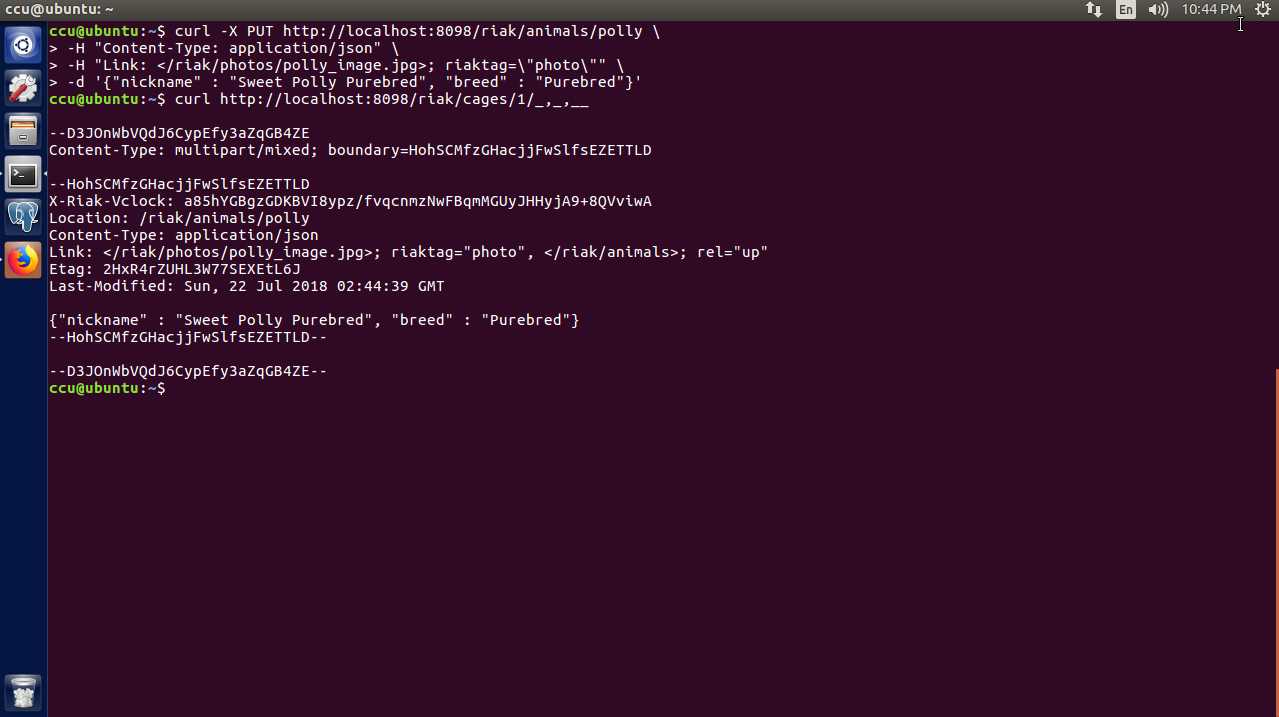
3. Read the example Riak config dev/dev1/etc/app.config, and compare it to the other dev configurations.



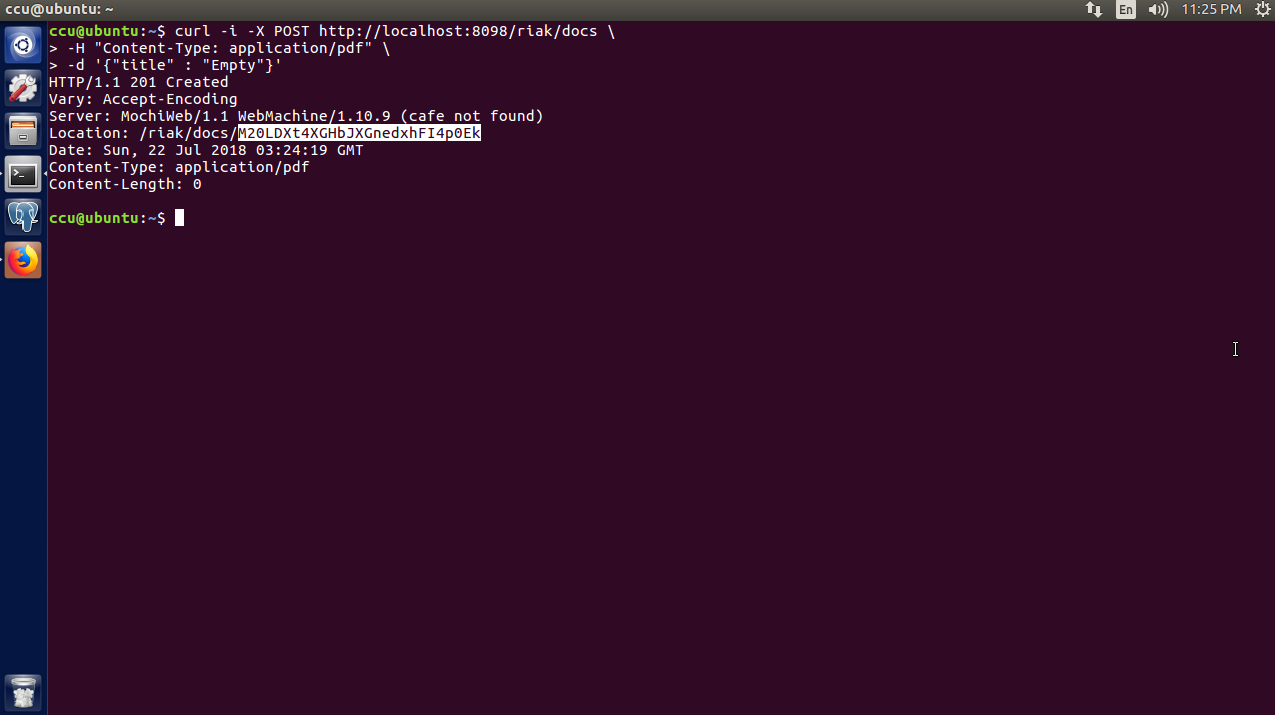
Since we are only using one instance of Riak, **/dev/dev1/etc/app.config** doesn’t exist. Instead, the singular Riak config is located at **/etc/riak/riak.config**.

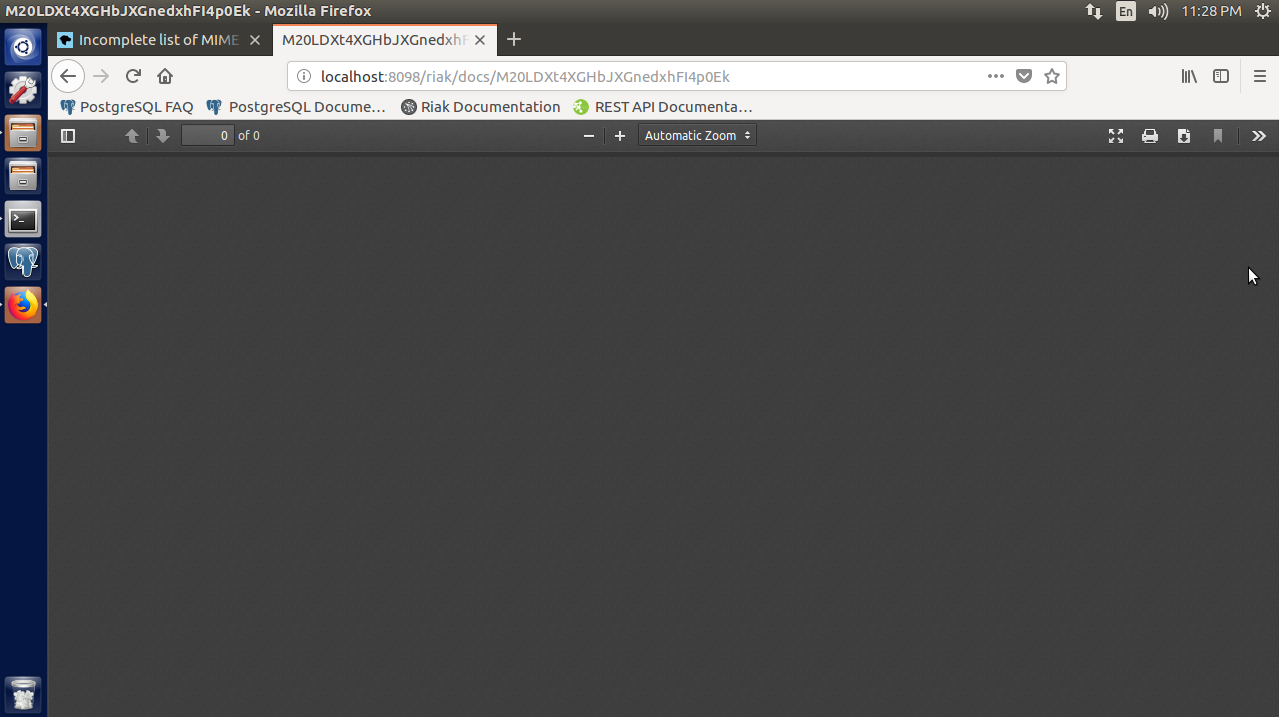
Do:

1. Using PUT, update animals/polly to have a Link pointing to photos/polly.jpg.



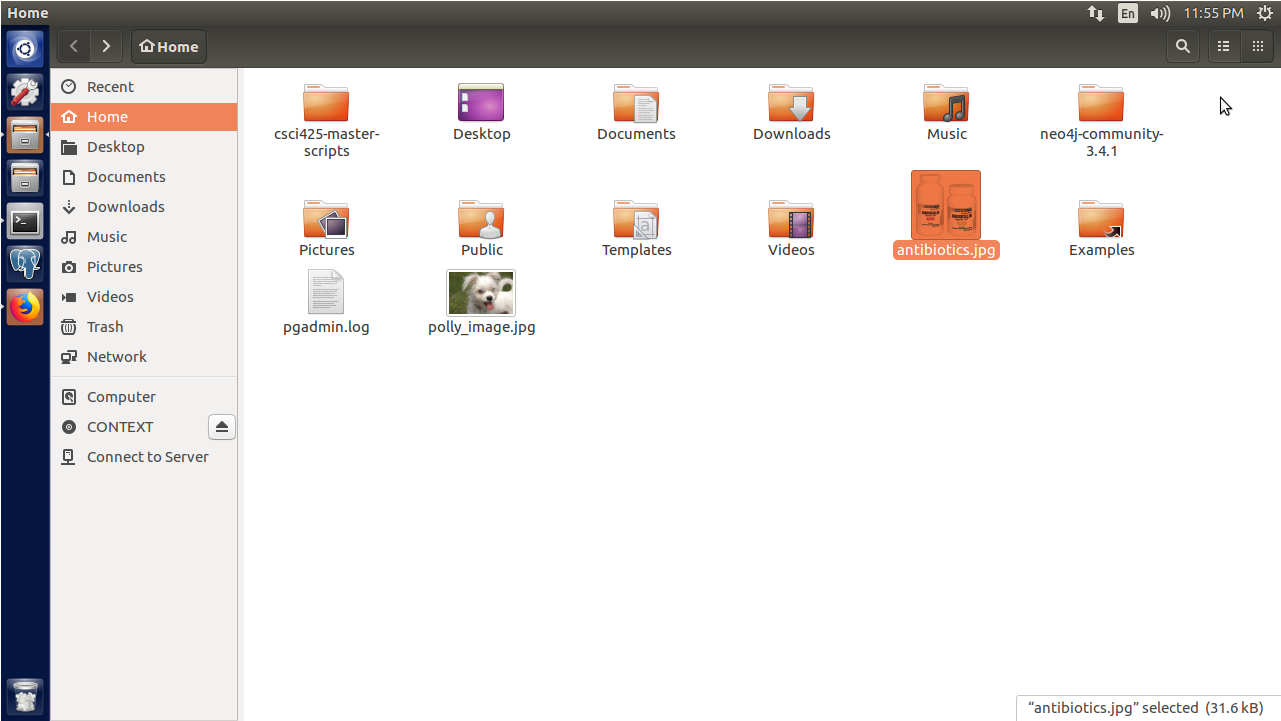
To update a key, you have to use the **PUT** command as if you were creating the key but adding the new information. In this case, I added a link from the animal named Polly to her picture in the **/riak/photos** directory. Unfortunately, it seems that using the entire **PUT** command is the only way to do this so far.

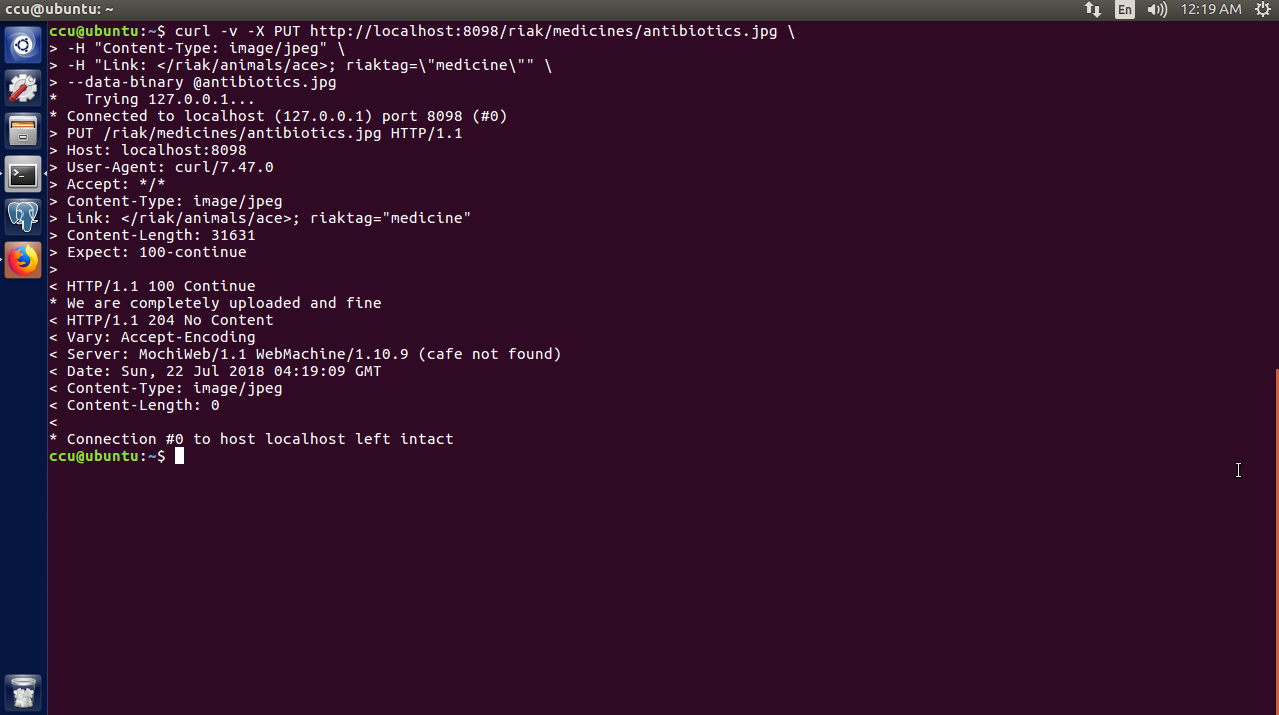
2. POST a file of a MIME type we haven’t tried (such as application/pdf), find the generated key, and hit that URL from a web browser.

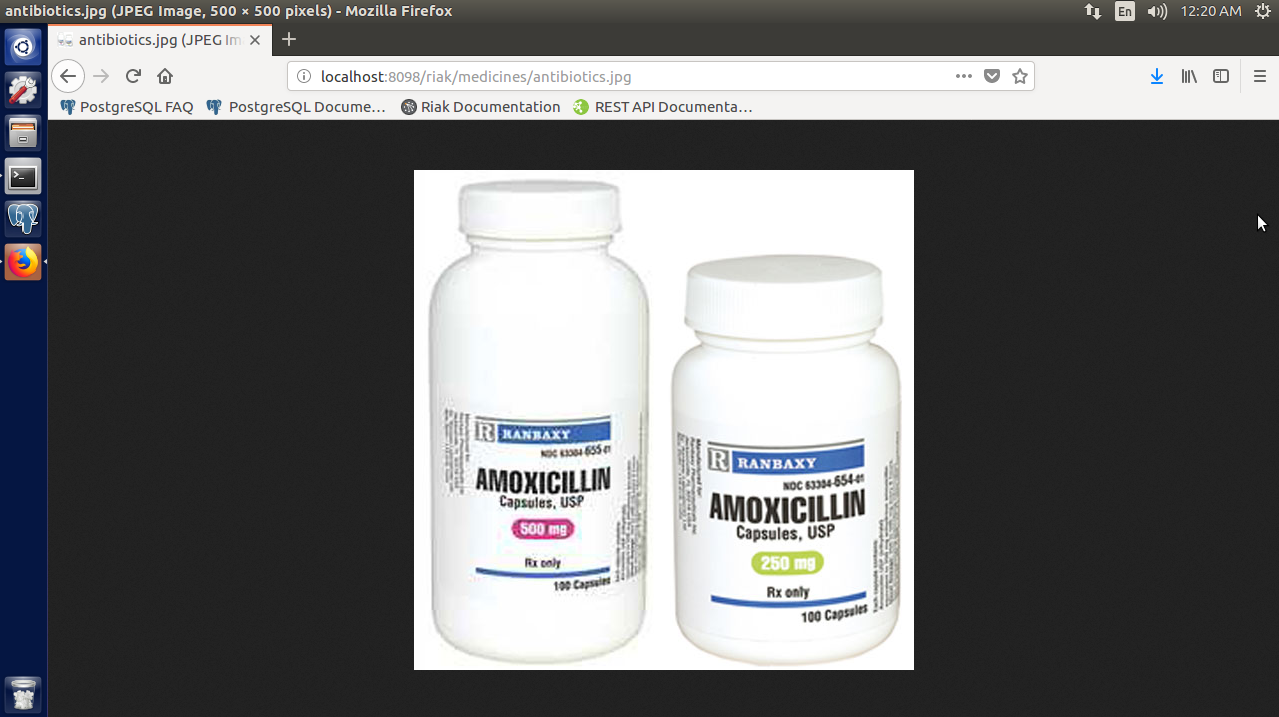


I decided to **POST** a **.pdf** file, so first I had to look up the MIME type, which was **application/pdf**. I used the **-i** option for **POST** to see the generated key once it was created. I had created a new bucket for the new key called **docs**, so I inspected the pdf at the address in the screenshot and a pdf viewer was displayed. The pdf was blank, but since the viewer showed up in the browser I assume it was created successfully.

3. Create a new bucket type called medicines, PUT a JPEG image value (with the proper MIME type) keyed as antibiotics, and link to the animal Ace (poor, sick puppy).





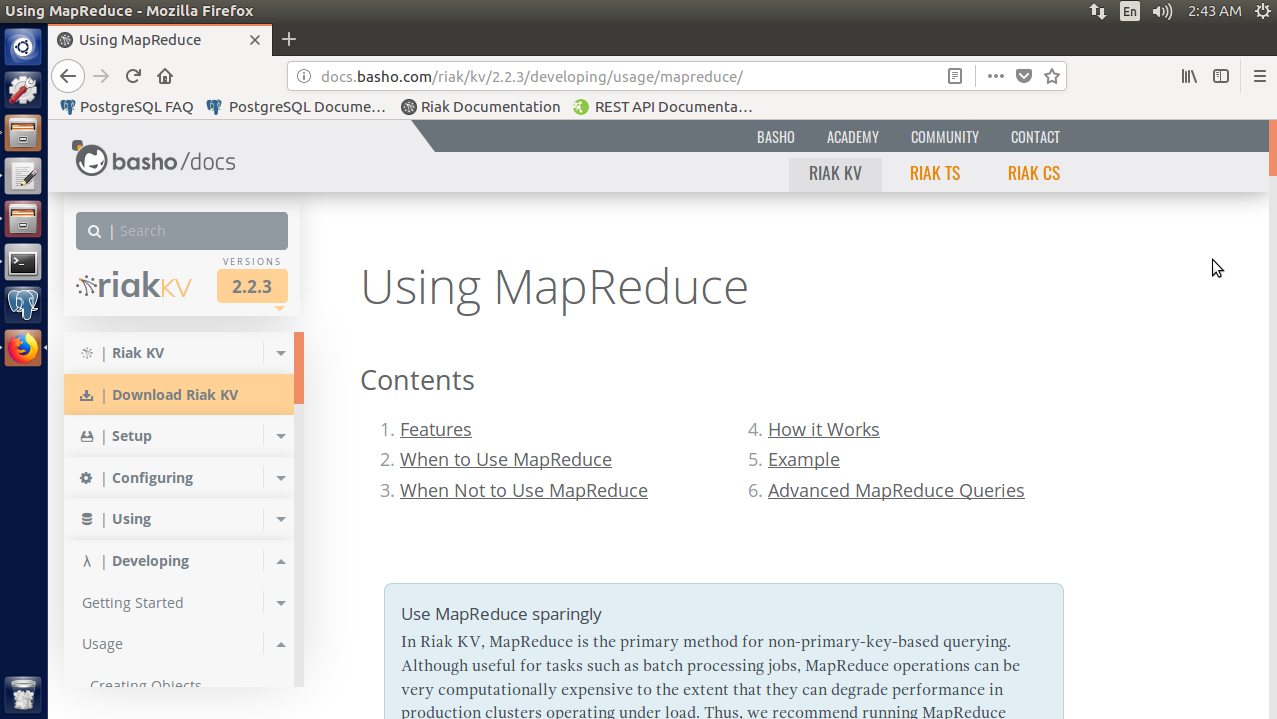


First, I put a random picture of dog medicine into the home directory. Then, I used the **PUT** command to create the new bucket, **medicine**, put the new key inside of it and link it to the **ace** key. Finally, I went to the new key’s address in a browser to make sure it was there, and it was.

**Day 2**

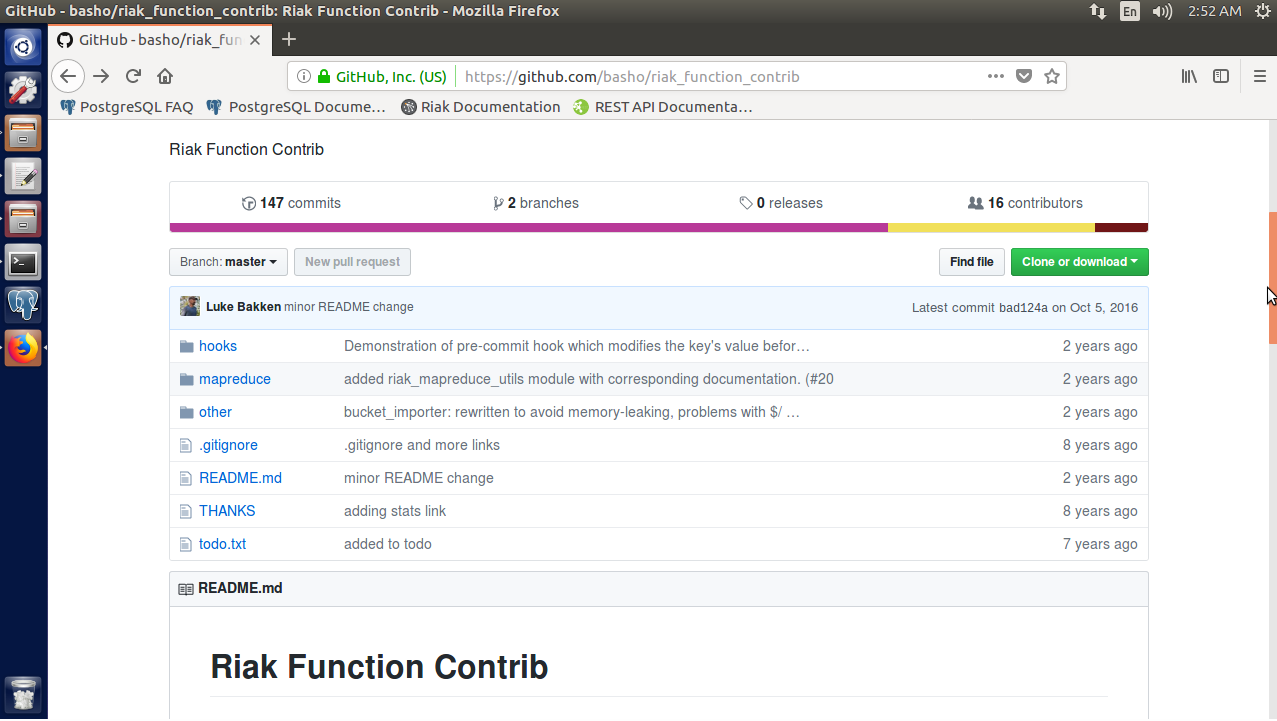
Find:

1. Read the online Riak mapreduce documentation.



To find the **mapreduce** documentation, I went to the Riak documentation page and searched for it.

2. Find the Riak contrib fuctions repository, with lots of prebuilt mapreduce functions.



Through a Google search, I found that the Riak contrib repository is hosted on GitHub. The **mapreduce** contrib packages are linked to on that page.

3. Find the online documentation for a complete list of key filters, which range from converting strings to\_upper to finding numerical values between some range to even some simple Levenshtein distance string matches and logical and/or/not operations.

Do:

1. Write map and reduce functions against the rooms bucket to find the total guest capacity per floor.

2. Extend the previous function with a filter to find the capacities only for rooms on floors 42 and 43.