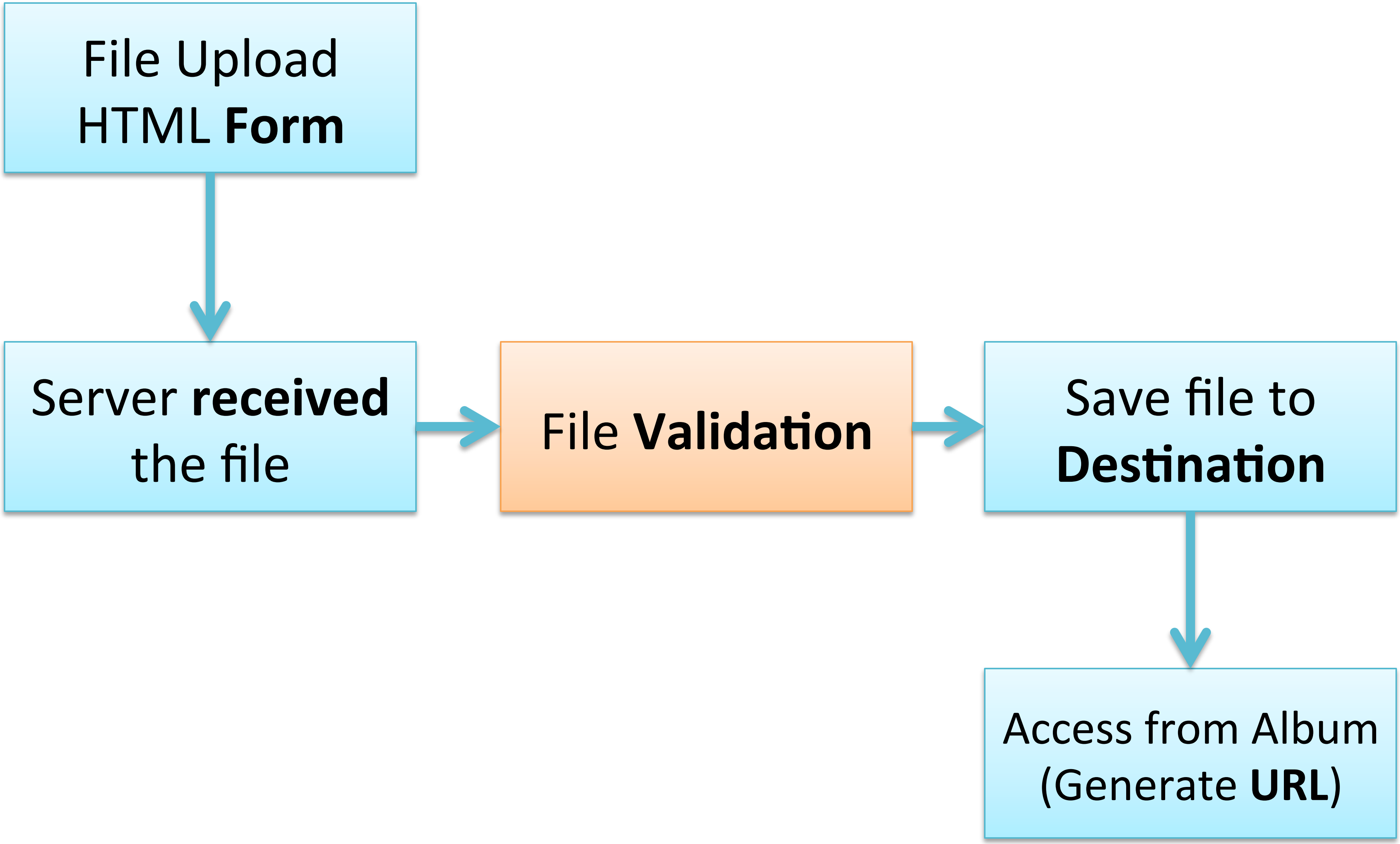
CSCI4140
Open-Source Software Project Development

Tutorial 3

Photo Upload: Part (2)
DBI module in Perl
Debugging in Perl

Last Update: 4 Feb 2014

Photo Upload – Overview



File Upload HTML **Form**

Server **received** the file

File **Validation**

Save file to **Destination**

Access from Album (Generate **URL**)

Photo Upload – Server Side Validation

- Is file too large ?
  - Check script in previous tutorial

- Is the file really an image ? / Is image extension match its type ?
  - Check using command line utilities

- Any file with same name exist ? / Duplicated ?
  - Using perl operator
  - Trigger duplication handling interface

ImageMagick

Command Exec.

- ImageMagick
  - Command line tools for image edit, convert, …
  - Useful for checking image format and thumbnail generation

- Available on OpenShift Perl cartridge
  - Also on department `linux*`
- Install on your own computer …
  - Ubuntu: `sudo apt-get install imagemagick`
  - Mac: `brew install imagemagick`

Official Website: http://www.imagemagick.org/

ImageMagick

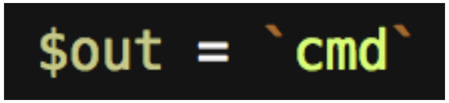
Command Exec.

- `identify <filename>`
  - Give **format** and characteristics of image file
  - Use the returned string to check correct image format
    - Containing some keywords ?
    - Regular expression

- `convert <in_file> <out_file>`
  - Convert the image
    - Control image type by `out_file` extension
    - More flags to control output style

Execute Command in Perl

- Commands can be executed using **backtick** in Perl



```
$out = `cmd`
```

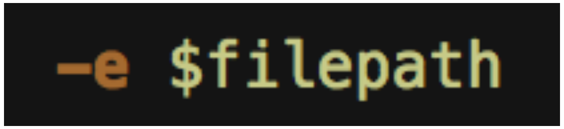
  - cmd is just a string
  - Variables inside will be interpolated

- `stdout` of command execution is returned
  - Use regular expression to filter output

More: http://perldoc.perl.org/perlop.html#Quote-Like-Operators

Image Upload

- Image Duplication
  - Check existence of a file by –e statement



```
-e $filepath
```

- If file already exists …
  - Trigger file duplication handling interface

Image Upload

Duplication

- File duplication handling interface
  - Allow user to **change filename**
    - Do not let user to change file extension
  - **Overwrite** existing file
  - **Discard** uploaded file

Image Upload

Duplication

- You have to save file once you received
  - Or the file will lost after running your script

- If file with same filename exists …
  - Original file being overwritten

- So, how to rename a duplicated file in duplication handling ?

DATABASE

Perl: DBI Module

- DBI: Database Interface
- Handle connection to database

- Consistent database interface for different DBMS
- Database Driver (DBD) is specific to DBMS type
  - You don't need to care about this level

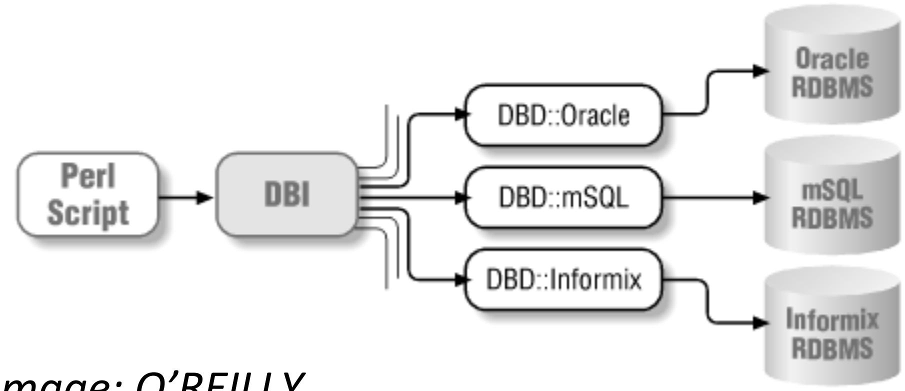


Image: O'REILLY

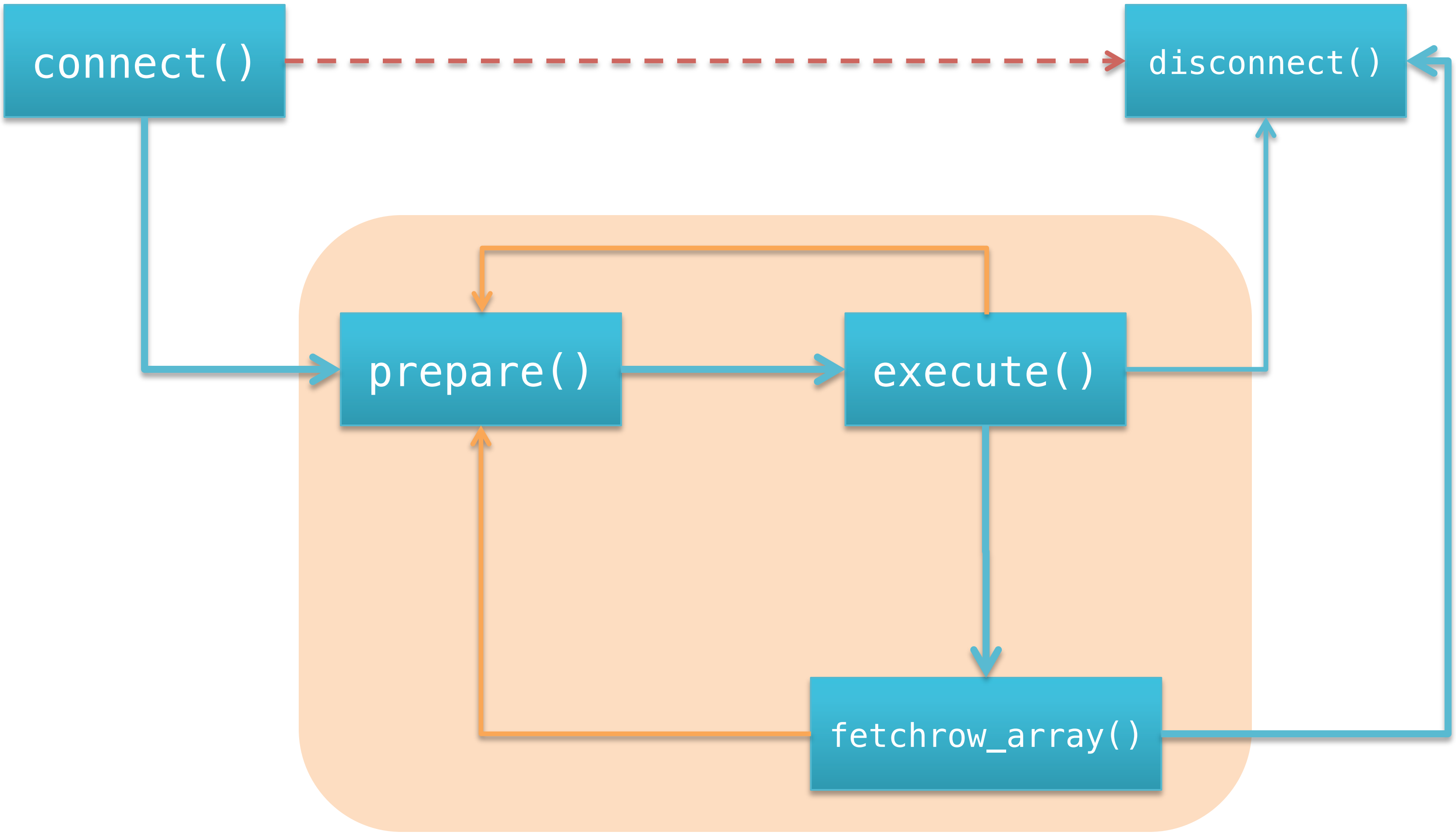
OpenShift: Database Setting

- Add MySQL cartridge to you application first
- Use environment variables retrieve database information

`OPENSHIFT_MYSQL_DB_HOST`	**Hostname** of database
`OPENSHIFT_MYSQL_DB_PORT`	**Port** of database
`OPENSHIFT_MYSQL_DB_USERNAME`	**Login** credential to database
`OPENSHIFT_MYSQL_DB_PASSWORD`	**Password** to log into database

- Default **database name** is same as application name
  – Use environment variable `OPENSHFIT_APP_NAME`
  – Or you can create another database …

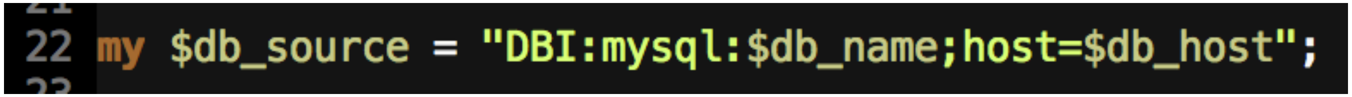
Perl DBI



More: http://search.cpan.org/~timb/DBI-1.616/DBI.pm#

Perl DBI: Connection to Database

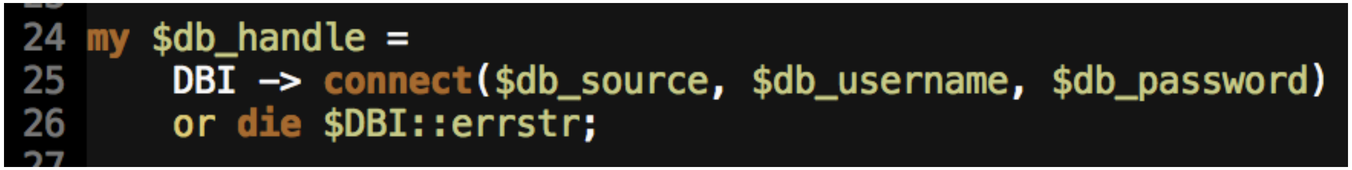
- Generate the database source string



```
my $db_source = "DBI:mysql:$db_name;host=$db_host";
```

  - Specify database type : database name; host

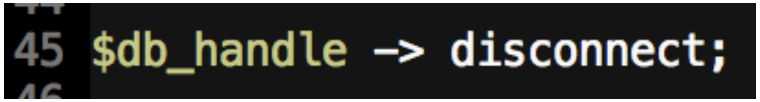
- Connection to database



```
my $db_handle =
    DBI -> connect($db_source, $db_username, $db_password)
    or die $DBI::errstr;
```

  - If connection failed, statement after or is executed

- Remember to disconnect at the end

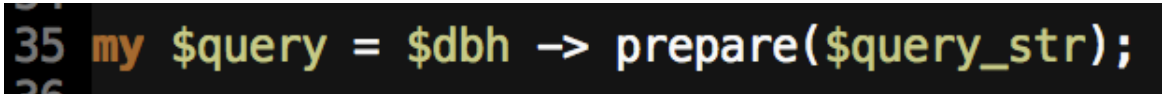


```
$db_handle -> disconnect;
```

tutorial3/db_connect.cgi

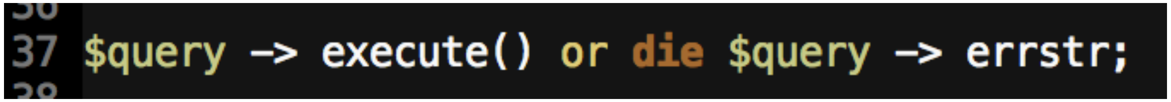
Perl DBI: Query

- Write the SQL query
- Prepare a query



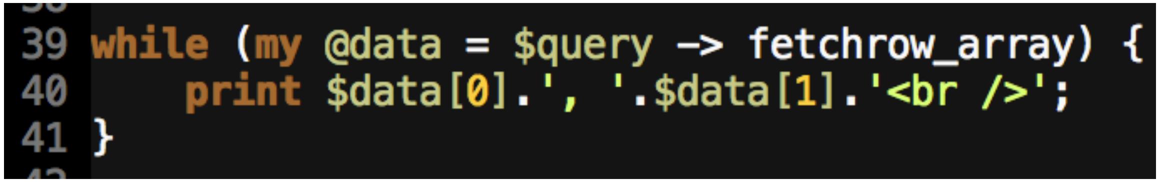
```
my $query = $dbh -> prepare($query_str);
```

- Execute



```
$query -> execute() or die $query -> errstr;
```

- Get result of query



```
while (my @data = $query -> fetchrow_array) {
    print $data[0].', '.$data[1].'<br />';
}
```

`tutorial3/db_query.cgi`

Database

- What to store in database?
  - Uploaded photos attributes
  - Session
  - User credential … ?
- Design your own schema
- Write SQL query
  - Or get query from phpMyAdmin

phpMyAdmin

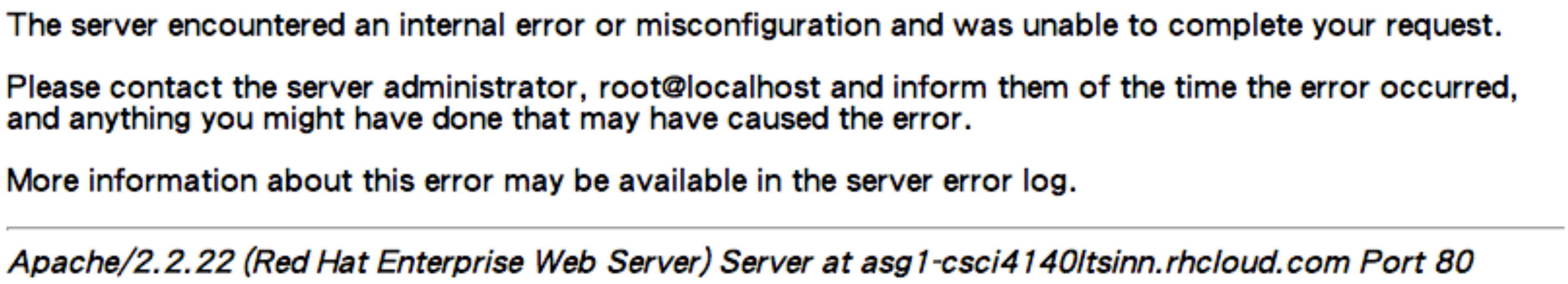
- Handy database management interface
  - UI to manage, view, edit database table / entry
  - Show you SQL query

- OpenShift provide cartridge
  - Add cartridge to application
  - Access from following URL:

    `http://[appname]-[domain].rhcloud.com/phpmyadmin`
  - Username and password can be found in application overview

DEBUGGING

Internal Server Error



The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, root@localhost and inform them of the time the error occurred, and anything you might have done that may have caused the error.

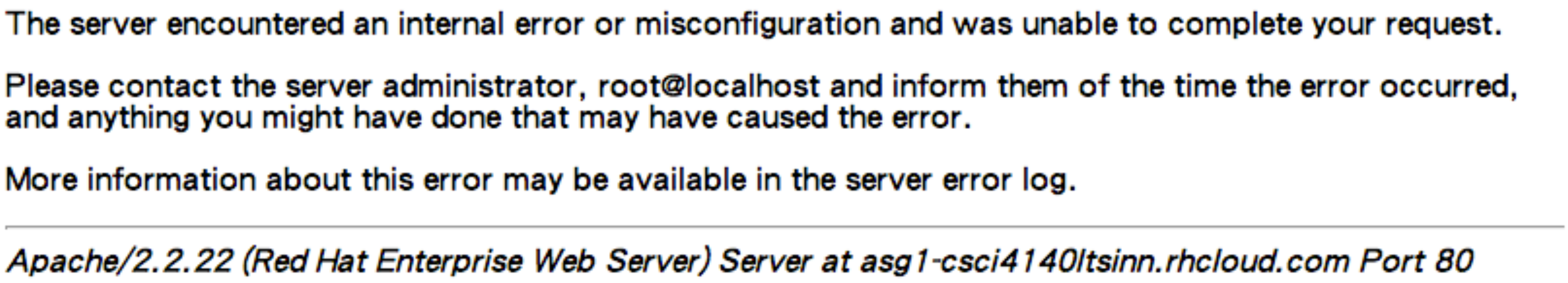
More information about this error may be available in the server error log.

Apache/2.2.22 (Red Hat Enterprise Web Server) Server at asg1-csci4140ltsinn.rhcloud.com Port 80

- May due to **bug** in script, fail to **execute**, or even **syntax** error
  - – No error detail is expose to user

- How can we debug ?

Internal Server Error



The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, root@localhost and inform them of the time the error occurred, and anything you might have done that may have caused the error.

More information about this error may be available in the server error log.

Apache/2.2.22 (Red Hat Enterprise Web Server) Server at asg1-csci4140ltsinn.rhcloud.com Port 80

- SSH to OpenShift and Apache log
- CGI module
- Perl Data Dumper

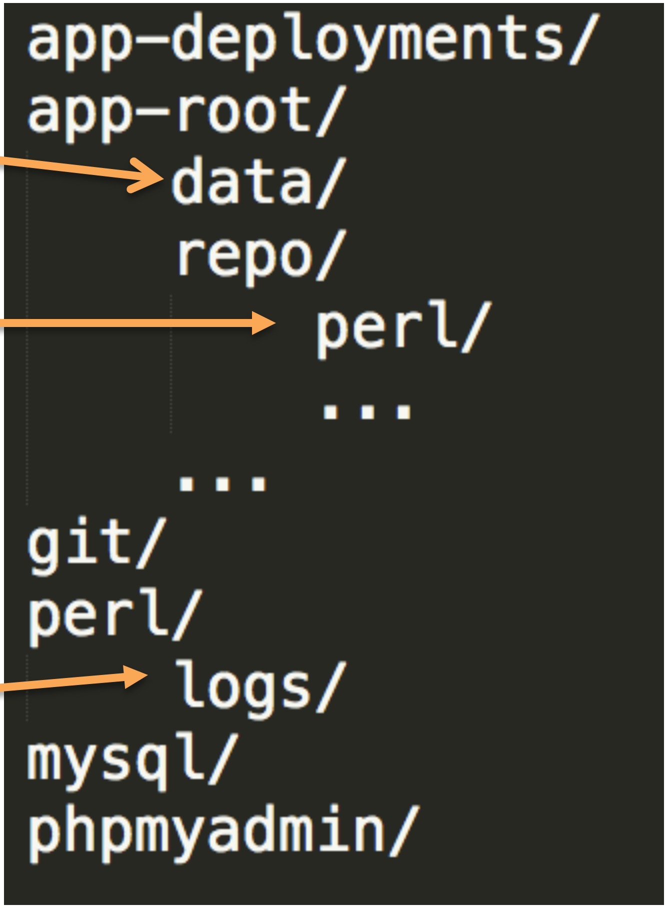
SSH Access to OpenShift

- OpenShift server can be access via `ssh` directly
  - Check file upload and save operations
  - Minor change to code for simple bugfix or testing
    - Your change will be flush in next deployment
    - Copy you modification before next push!
  - Access log for error (bug) tracing

- Host and login information ?
  - SSH Config
    - From URL of `git clone`
    - Refer to tutorial 1 (OpenShift)

SSH Access to OpenShift

- Directory structure (extract)
  - Persistent directory
  - Current deployed version
    - Changes will reflect immediately here
  - Apache logs can be found here



```
app-deployments/
app-root/
    data/
    repo/
        perl/
        ...
    ...
git/
perl/
    logs/
mysql/
phpmyadmin/
```

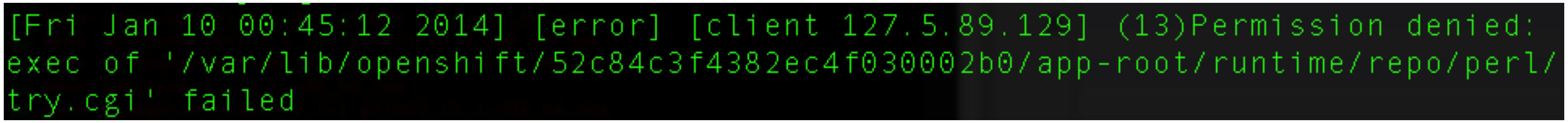
Apache log

- Access log
  - All received requests
  - Not our main concern

- **Error log**
  - Our main focus
  - Log all error with timestamp
    - Check current time with `date` command

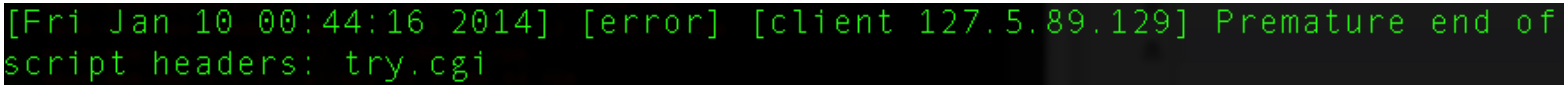
Apache log

- Error Log: Example
  - Permission Denied



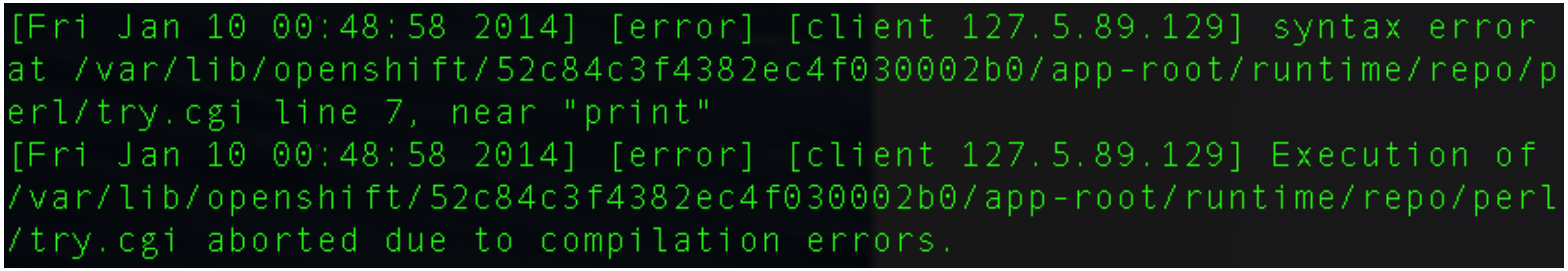
```
[Fri Jan 10 00:45:12 2014] [error] [client 127.5.89.129] (13)Permission denied:
exec of '/var/lib/openshift/52c84c3f4382ec4f030002b0/app-root/runtime/repo/perl/
try.cgi' failed
```

    - Did you enable execute of script ? (chmod a+x)
  - Premature end of script header ?



```
[Fri Jan 10 00:44:16 2014] [error] [client 127.5.89.129] Premature end of
script headers: try.cgi
```

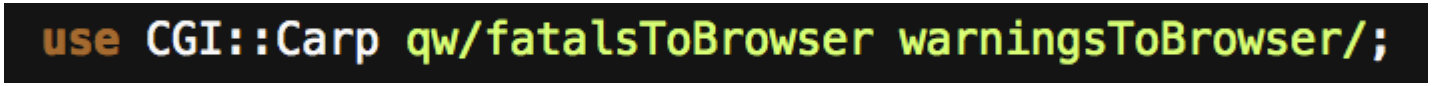
    - Did you print HTTP header ?
  - Syntax Error …



```
[Fri Jan 10 00:48:58 2014] [error] [client 127.5.89.129] syntax error
at /var/lib/openshift/52c84c3f4382ec4f030002b0/app-root/runtime/repo/p
erl/try.cgi line 7, near "print"
[Fri Jan 10 00:48:58 2014] [error] [client 127.5.89.129] Execution of
/var/lib/openshift/52c84c3f4382ec4f030002b0/app-root/runtime/repo/perl
/try.cgi aborted due to compilation errors.
```

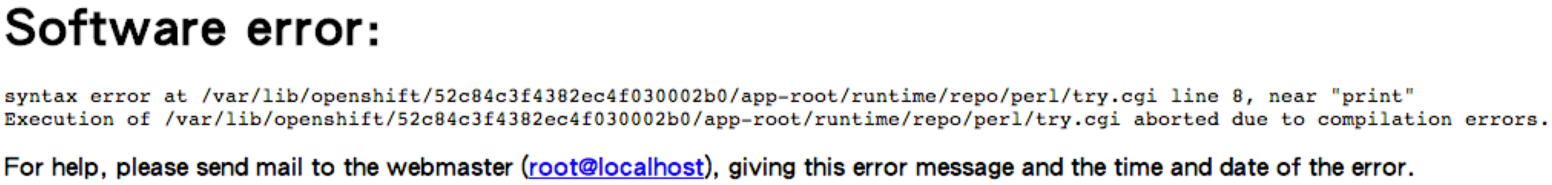
Print Error Message to Browser

- Add the following line to beginning of your script



```
use CGI::Carp qw/fatalsToBrowser warningsToBrowser/;
```

- Produce more detail output
  - Example, a syntax error:



Software error:

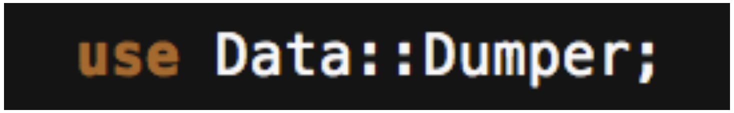
```
syntax error at /var/lib/openshift/52c84c3f4382ec4f030002b0/app-root/runtime/repo/perl/try.cgi line 8, near "print"
Execution of /var/lib/openshift/52c84c3f4382ec4f030002b0/app-root/runtime/repo/perl/try.cgi aborted due to compilation errors.
```

For help, please send mail to the webmaster (root@localhost), giving this error message and the time and date of the error.

Perl Data Dumper

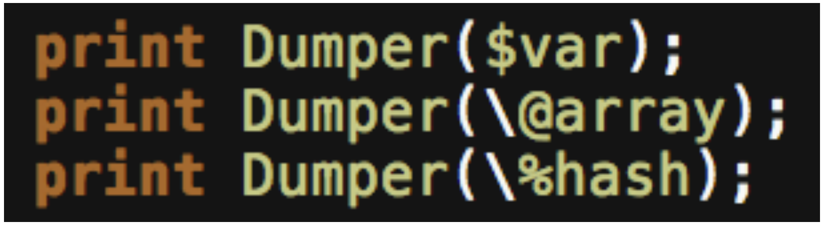
- Dump content of variable
  - Similar to `var_dump()` in php

- Declare use of Data Dumper



```
use Data::Dumper;
```

- Dump result as following:



```
print Dumper($var);
print Dumper(\@array);
print Dumper(\%hash);
```

`tutorial3/dumper.cgi`

END

Bonus: Timezone of Database

- OpenShift server default timezone is NOT GMT+8

- If you use `CURRENT_TIMESTAMP` as upload time, displayed time will not match your clock

- Change timezone of MySQL server by

```
    SET GLOBAL time_zone = '+8:00';
```

  – Execute from phpMyAdmin