System Deployment & Testing

Due: Week 13 – 02/11/2018 3:00PM

By week 13 your application will be running in a production environment in "the cloud" and your application will have at least one functional unit-test.

Deployment Environment Specifications

Each group will setup access to Amazon Web Services (either provided by signing up a free trial or through AWS Educate). You must create a Linux instance on AWS which will host your website. Do not expect a graphical environment - you will need to configure this server from the command line. As your team will be required to implement your web application on this server, you should ensure your web application can run under following specifications (these are of an AWS t2.micro instance).

- Ubuntu Server 14.04 LTS
- 1 Virtual CPU (Intel Xeon)
- 1 GiB RAM
- 8 GiB Storage (EBS)

You are free to create any type of Linux instance on AWS, however be weary of how much free credit you will be allocated. If you exceed the free credit available, you will need to pay for the services yourself.

Deployment Requirements

You will need to implement the following requirements in your production environment.

- **Reverse Proxy (nginx):** Requests from the internet to your web application should hit a reverse proxy, which will forward appropriate packets to your web application.
  You must reduce server load by configuring nginx to serve static files (i.e. JS, CSS, images, media, etc.) instead executing your application code.
- **Fault Tolerance:** If your server reboots, it should not need human intervention to begin serving requests as normal again.

Unit-testing

Provide unit tests for at least five different pieces of functionality, making a reasonable attempt at covering a significant portion of your web application's back-end.
If your framework has support for unit-testing (as Django does), implement it in your frameworks style.

Security

Your server, database and AWS account must be configured with appropriate security precautions and restrictions in place.

Documentation

You must provide some developer documentation to describe how the deployment environment is constructed. This should aid any future developers in uncovering how your system operates.

You may incorporate this documentation into your final System Delivery documentation.

System Deployment & Testing continued on next page. . .

Access for Markers

You must provide markers access to your AWS interface and your Linux server instance.
To do this, you *must* configure the following components:

- **IAM User**

  An IAM user must be configured with ‘ReadOnlyAccess’ to your AWS account. This will allow the markers to access and assess your AWS set-up. Include the ‘IAM Users Sign-In Link’ and credentials in your Week 13 documentation.

  Following are basic instructions on creating an IAM user with ‘ReadOnlyAccess’:

  1. Log into the AWS Management Console.
  2. Open the ‘Identity and Access Management’ (IAM) service.
  3. Take note of the ’IAM Users Sign-in Link’.
  4. Under ‘Details’, click ‘Users’.
  5. Click ‘Create New Users’
  6. In the ‘1.’ field, enter a username for the marking account.
  7. Click Create.
  8. Click Close (there is no need to download the credentials).
  9. Back in the IAM section, in Users click on the new user you created.
  10. Select the ‘Security Credentials’ tab.
  11. Click ‘Manage Password’.
  12. Select ‘Assign a Custom Password’.
  13. Enter and confirm a new password for marking access.
  14. Click ‘Apply’.
  15. Select the ‘Permissions’ tab.
  16. Click ‘Attach Policy’.
  17. Select the ‘ReadOnlyAccess’ checkbox.
  18. Click ‘Attach Policy’
  19. Test the account: browse to the ‘IAM Users Sign-in Link’ and ensure the new user has read-only access to all aspects of your web application.

- **SSH Access**

  You must append the following SSH public key to the file: `/home/ubuntu/.ssh/authorized_keys`. This will allow markers to log into your Linux instance and assess it’s configuration.

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQC96fmaBKrsFurlr39PxdUhDze4M43a
9WzEGPPBBgMmm48WmvujIvSyqrS/qMpaOz+wPSFSduSiA47X/anjG0xFRrLcBVCLrlvmmFxtml73kyGeTUQ
EUYKbMXEi+TbfoXkUl+oDuj973UUpUIqyJzUSZArKKwZWZH1iqLMlOajeLu55vZTxnmgREFt96CxZxCNau1
IWkMe0Y1QPx75RLJnvqX5u2sIV4l0p9plMq7z3WHwPKvp673tM0nBzNGsDTKSGbu5EjpaQWuZ5pW7Vb66Cp
LeUk1047xr5T2+uF1uQnbfjxgZEPBYtrt0sHHbb1cWrGRfJQ2zFXsUOaVEJKtrLHnr4gYqtdQPGUYwaa490
TB0DjC65xmY8aMT6PHrWvLvZhd7SSten7c7nEtduGvLQXL/P8xRlB4FYVAtgt1jH2R64sLC6wPeXu2x4mrK/
1w802cgY3l3gZTPT2SnHmab3+vWVF3a471GphD4jdWjJX+ITv3v9J/gdFTgfZKDKqgJzX6d3K982+RjVftBD
Tu/us8HUME4cXw6cIiJl6rYAXYeMJkvId+op2hXWv12uCUCXw/RR03Y85TbQ1akgJbN943ARiJvTuZc4kJ9U
MfG6bHuByZJZCJ2zNczejlb4myh9LXakFrSjMZGz6T2Co8gcfchwatNFrVAqhCP0B9T+bw== markers@elecx609.com
```

  (ensure that the line-breaks are removed and that this all appears on a single line)

- **Database Access**

  You must ensure your application server has a functioning terminal-based frontend to your database (eg, `psql` or `mysql`) to allow markers to review your database state and structure.

In your documentation, provide credentials and basic instructions for markers to obtain terminal-based access to your database from your application server.

Deploying to production always takes longer than expected.
Do not leave deployment until Week 13 - you must have your production environment running (in a basic capacity) several weeks prior.

Marking Criteria

	Novice	**Competent**	**Proficient**
System Set-up	**0-1:** Instructions have not been followed, server is poorly configured or does not work as it should.	**2-4:** Most aspects of the setup have been followed, but there is room for improvement or some configurations are incorrect.	**5-6:** All specified configurations have been incorporated and the student has clearly invested effort into configuring their server properly and securely.
Unit Tests	**0-2:** Insufficient unit testing or unit tests do not cover enough features.	**3-4:** Unit tests are decent, but don't add considerable value or don't test the most important features.	**5:** Sufficient unit testing with appropriate coverage of website features.
Security	**0-1:** Significant vulnerabilities found.	**2-3:** Minor security issues identified, but would not cause full system compromise.	**4:** No security issues identified, system appears to be secure.