# ICT & Infra S3 Automation & Orchestration, week 3

|  |  |
| --- | --- |
| Class: | I02 |
| Student number: | 4538528 |
| Student name: | Mihai Glodici |

## Introduction

This week you will practice creating Ansible Playbook(s) to automate a complex process. Before executing the assignment, ensure that you have working Ansible control node. For this assignment, you must be familiar with preparing Apache server and hosting Flask application on it. To repeat the necessary steps, you can read this simple tutorial, which explains Apache installation steps; creation of simple Flask application; hosting a Flask app on Apache: <https://www.codementor.io/@abhishake/minimal-apache-configuration-for-deploying-a-flask-app-ubuntu-18-04-phu50a7ft>

Similar (probably the same) steps you will need to automate in this assignment.

### Assignment 1. Create a Playbook to automate hosting of a simple Flask application

### Difficulty: ★★★★☆.

You already know how to create a simple Ansible Playbook. Now it is time to create a Playbook that can prepare newly installed Linux machine to host a website. Additionally, you should be able to host Flask application automatically. For this assignment, you must execute the following steps:

1. Using Amazon EC2 service, manually create and configure (free tier) Ubuntu instance to host a website.
2. Test SSH connection to the Virtual Machine.
3. Read the steps (from codementor.io tutorial) of Apache configuration and simple Flask app hosting.
4. Create a simple Flask app (my\_flask\_app.py) on your local machine.
5. Create a Playbook to **automate**:
   1. apache2 service installation on the VM;
   2. mod\_wsgi module installation;
   3. Flask installation;
   4. Generate and upload “my\_flask\_app.wsgi” file using Jinja2 templating. HOW TO: [Template a file out to a remote server](https://docs.ansible.com/ansible/latest/modules/template_module.html)
   5. Generate and upload “ExampleFlask.conf” file using Jinja2 templating.
   6. Check if newly created/uploaded website is accessible. HOW TO: [Interacts with webservices](https://docs.ansible.com/ansible/latest/modules/uri_module.html)
6. Manually test if the website is working.

Provide screenshots (evidence) for your solution. Always explain your evidence! As a prof, we expect at least:

* A screenshot of running EC2 Ubuntu instance;
* A screenshot of successful SSH connection to the VM;
* Ansible Playbook file(s) that automates configuration of the VM.
* A prof that the website is working.

|  |
| --- |
| *Solution:* |

ssh -i "ubuntu2.pem" [ubuntu@ec2-3-68-91-213.eu-central-1.compute.amazonaws.com](mailto:ubuntu@ec2-3-68-91-213.eu-central-1.compute.amazonaws.com) command was used to connect through SSH

A picture containing background pattern

Description automatically generated

Successful playbook execution ^

Text

Description automatically generated

Ansible playbook .yml code

Graphical user interface, text, application, email

Description automatically generated

Website hosting throws a permission denied error after uploading our own website. Original Apache website posted correctly.