# ICT & Infra S3 Automation & Orchestration, week 3

|  |  |
| --- | --- |
| Class: | I3-cb01 |
| Student number: |  |
| Student name: | Joep Emons |

## Introduction

This week you will practice creating Ansible Playbook to automate a complex process. Before executing the assignment, ensure that you have working Ansible control node.

### Assignment 1. Ansible Playbook Development for Managing AWS EC2 Instances with Secrets Encryption and Debugging

### Difficulty: ★★★★☆.

The objective of this assignment is to assess your understanding of Ansible and your ability to create a playbook that includes multiple tasks, uses conditional statements, variables, Jinja2 templates, and the debug module ~~while securely managing secrets with Ansible Vault~~. To start with the assignment, first you must create one virtual instance running Ubuntu or Amazon Linux distributions.

**Requirements:**

###### 1. AWS EC2 Instance Creation:

Create a new AWS EC2 instance running either Ubuntu or Amazon Linux distributions. This instance should be accessible via SSH.

###### 2. Playbook Structure:

Create an Ansible playbook named manage\_ec2\_instances.yml.

###### 3. Variables:

Define some variables at the beginning of your playbook to make it easy to customize the configuration in the future.

###### 4. Conditional Statements:

Write conditional statements to detect the OS distribution of the AWS EC2 instances (Ubuntu or Amazon Linux). Based on the distribution, install the appropriate web server package (apache2 for Ubuntu and httpd for Amazon Linux).

###### 5. Task: Secure Debugging:

Create a task using the debug module to display information about the AWS EC2 instance, such as IP addresses, SSH user, installed web server package, and other non-sensitive information.

###### 6. Templating with Jinja2:

Create a Jinja2 template named index.html.j2 that includes a basic HTML page. Use placeholders within the template for the website title and content.

###### 7. Task: Configure AWS EC2 Instances:

Create a task to copy the index.html.j2 template to the appropriate web server document root directory on the AWS EC2 instances based on their OS distribution.

Use Jinja2 template rendering to replace placeholders with the values you specify in the playbook.

Ensure that your playbook runs successfully on your AWS EC2 instance. Additionally, test if a webpage is accessible from outside.

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

* Playbook from manage\_ec2\_instances.yml file
* Content of index.html.j2 file;
* A prof that the website is working.



I created a ec2 instance using aws cli

Afbeelding met tekst, schermopname, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

Then I put all necessary data in inventory file so public ip user route to private ssh key then I set some variables in the ec2 instance group I disable ssh host key verification and I specify thant ansible should use python 3 on the remote host

Afbeelding met tekst, schermopname, violet

Door AI gegenereerde inhoud is mogelijk onjuist.

This is my playbook this installs apache2 and httpd on ubuntu then displays ec2 instance details I used this because I had some problems and AI recommended this for debugging then it creates a web page whit the jinja2 template ass its source and the template uses variables defined earlier to fill in content

Afbeelding met tekst, schermopname, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

A simple jinja2 template that shows server information

Afbeelding met tekst, schermopname, Lettertype, nummer

Door AI gegenereerde inhoud is mogelijk onjuist.  
proof that website can be accessed through public ip and is running apache2 webserver

|  |
| --- |
| Solution: what I did for this assignment   * Created ec2 instance whit AWS CLI * Added public ip and ssh key to inventory file so playbook can access ec2 * Created and configure appache webserver whit http * Used jinja2 templating like assignment access   Now the website is running in a ec2 instance that’s hosting a apache2 website managed by ansible |