### Hands-on Cloud Computing Services Lezione 3

Gabriele Russo Russo University of Rome Tor Vergata, Italy

A.A. 2021/22



- EC2 with ELB, VPC and Auto Scaling
- Application deployment through custom AMIs or cloud-init scripts
- What if the application (or its configuration) must be updated at some point?

# **IT Automation using Ansible**

- Ansible delivers simple IT automation that ends repetitive tasks and frees up teams for more strategic work.
- Agentless
- Define WHAT you want to achieve, instead of HOW
- https://www.ansible.com/
- Available for Linux and macOS: https://docs.ansible.com/ansible/latest/installation\_ guide/intro\_installation.html
- Windows users need a Linux-based VM
- Alternatives: Chef, Puppet, ...

## **Ansible: Key Concepts**

- Tasks and Playbooks
- Modules (e.g., file, archive, apt)
  - Built-in modules
  - Custom modules
- Inventory = hosts to be managed
  - Static
  - Dynamic

## A playbook for Photogallery: inventory

- Create the inventory file 'hosts.ini'
  - You may also put your local host in the inventory...)
- One line per host
- Possibly organized into groups (e.g., web, db, ...)
- We can add params for SSH authentication

### Inventory file

```
[web]
18.185.19.141 ansible_user='ec2-user' \
    ansible_ssh_private_key_file='/path/to/keypair.pem'
```

Simple test using the ping module:

\$ ansible -i hosts.ini -m ping all

# A playbook for Photogallery

To deploy Photogallery we need to:

- Upload application files (module: copy)
- Install dependencies (modules: yum, pip)
- Install systemd unit file to start server at boot (module: copy)
- Enable systemd service (module: systemd)

### Check deploy\_gallery.yaml

\$ ansible-playbook -v -i hosts.ini deploy\_gallery.yaml
# What happens if we try again?
\$ ansible-playbook -v -i hosts.ini deploy gallery.yaml

# **Ansible: Dynamic Inventory**

- Ansible requires an inventory
- Not necessarily a static file
- AWS Inventory Source: run your playbooks using (a subset of) your EC2 instances as target hosts (e.g., filtered by tag)
- Requires Ansible 2.9+
- A plugin required, easy to install:

\$ ansible-galaxy collection install amazon.aws

## **Ansible: AWS Dynamic Inventory**

Create a YAML file (name MUST end with aws\_ec2.(yml|yaml) → galleryInventory.aws\_ec2.yaml

#### Test

ansible-inventory -i galleryInventory.aws\_ec2.yaml --graph

### Running the playbook

```
ansible-playbook -i galleryInventory.aws_ec2.yaml
--private-key=path/to/key.pem -u ec2-user
deploy_gallery.yaml
```

# Ansible: More Advanced Stuff

- Groups and Roles
- Templates
- Ansible Tower / AWX<sup>1</sup>
  - Share playbooks / delegate
  - Schedule workflows
  - Dashboards

<sup>&</sup>lt;sup>1</sup>https://github.com/ansible/awx

### Amazon S3

- Scalable object storage service
- Pricing: https://aws.amazon.com/it/s3/pricing/
- Buckets and objects
- Let's create a bucket using S3 console
- Bucket name must be unique across all AWS regions and accounts
- We can choose who can access objects and buckets: https://docs.aws.amazon.com/it\_it/AmazonS3/latest/dev/ example-bucket-policies.html
- For Photogallery, we want everyone to read objects

We can reference an object like this:

https://BUCKETNAME.s3.amazonaws.com/FILENAME

## Using S3 through the CLI

\$ aws s3 ls \$ aws s3 ls s3://mybucket \$ aws s3 cp prova.txt s3://mybucket/ \$ aws s3 ls s3://mybucket \$ aws s3 rm s3://mybucket/prova.txt

Third-party clients also available: e.g., s3cmd

## **Hosting Static Web Content on S3**

- Objects in a public bucket can be accessed through HTTP
- You can use S3 to host static web content
  - a static website
  - the frontend of a web application
- To enable web hosting on a bucket: https://docs.aws.amazon. com/AmazonS3/latest/userguide/EnableWebsiteHosting.html

- Boto is the AWS SDK for Python. It enables Python developers to create, configure, and manage AWS services, such as EC2 and S3. Boto provides an easy to use, object-oriented API, as well as low-level access to AWS services.
- Similar APIs available for other languages as well
- We'll use boto3: https://boto3.amazonaws.com/v1/ documentation/api/latest/index.html
- We use boto3 to list objects in our bucket: s3list.py

# **Configuring boto3**

- Boto configuration is similar to AWS CLI configuration (default region, ...)
- Important issue: providing credentials to Boto
  - Especially important when deploying applications on remote machines
- Several ways to provide credentials (and configs): https://boto3.amazonaws.com/v1/documentation/api/ latest/guide/credentials.html
- If AWS CLI has been configured on your PC, boto3 works out-of-the-box

### Exercise

Extend PhotoGallery with the following features:

- display pictures stored in a S3 bucket, along with their upload time
- users can upload pictures

