Project Plan

Case Study: Cloud infrastructure orchestration

Group: RDEnvi

Group members:

* Robert Asvicas
* Dmitry Lvov

# Project background

We are a group of 2 students that specializes on making infrastructure solutions for software developers. In this case we will be teaming up with 2 software developing groups with a purpose of helping them host their products on our infrastructure. Working on AWS allows us to be very flexible which means that we will be able to satisfy their needs.

# Project Goal

Our main goal is to build an infrastructure which will be suitable for hosting certain software products created by developers in an advanced and automotive way. The infrastructure is going to be mainly based on AWS which will work in correlation with local Fontys servers which are planned to be used as a database. As foretold, our infrastructure will be automated and orchestrated via an advanced tool called Ansible. Lastly, our environment will follow the most progressive and advanced security innovations due to the implementation of stable logging mechanism, Software Defined Networking (SDN), Software Defined Wide Area Network (SD-WAN) and several monitoring tools similar to or Nagios itself.

# Problem Definition

# Milestones

|  |  |
| --- | --- |
| Deliverables | Expected delivery time |
| Project Plan | Week 3 |
| Design Document | Week 3 |
| Project Report | Week 3 |
| Amazon Web Services   * EC2 Web/App Servers * Control node (Ansible) * Main VPC for Servers   + Elastic Load Balancers for App and Web Servers   + Nat Instances   + 3 Availability zones   + Public and private subnets * Secure VPC for Databases * Transit gateway * S3 bucket for logs | Week 5  Week 4  Week 7-9  Week 7  Week 6  Week 6  Week 7  Week 8  Week 9  Week 9 |
| Fontys server   * Private database for storing keys | Week 9 |
| Playbooks for Ansible | Week 3-9 |
| Terraform code | Week 3-9 |

# Deliverables

* Configured infrastructure
  + Terraform code
    - AWS Policies
    - EC2 Instances
  + Ansible playbooks
    - SSH
    - Apache
    - Apache + Flask
    - Nginx
  + Virtual Private Cloud (Main)
    - 3 Availability Zones
    - Public Subnets
      * Nat Instances
    - Private Subnets
      * Web/App servers
      * Elastic Load Balancers
  + Virtual Private Cloud (DB)
    - 3 Availability Zones
    - Private Subnets
      * Database instances
  + Transit Gateway
* Documentation

# Expected results

* Fully functioning infrastructure that will be suitable for hosting software products
  + - * + AWS