## YD demo usecase

A customer asked you a demo of Cycloid.

Connect on Cycloid using the provided credentials https://console.cycloid.io

**Create a new project using the Magento stack.**

**User : pascal.gasp@gmail.com**

**Organisation ; cycloid-test-presales**

**Project :** cycloid-magento-demo

**Github :** <https://github.com/pgasp/cycloid-magento>

Since you are in a demo, you don't need to provide a valid configuration to run the pipeline.

* **The customer asks you if it's possible to add custom jobs in the pipeline.**

**Modify the pipeline to add a unit test job that execute this command : `echo 'testing source code ...'` before the build.**

**Describe what you changed into the pipeline configuration.**

In pipeline, I added new job uniTest starting with condition of git\_magento-code. That job will spin up a docker image to run a script.

I modified the build job to add as prerequired job UnitTest.

* **Then the customer is asking how could he modify the Magento stack to add a public Aws S3 bucket to store Magento's static files.**

We should add Job that will referrer to Terraform script to create and configure S3 bucket then create ansible script who is going to configure the magento instance to point to the S3 URL.

Here an example of Terraform script, I will use as baseline to set S3 bucket to serve web static files : <https://github.com/cycloidio/terraform-website-s3-cloudfront-route53/blob/master/site-main/main.tf>

Regarding the ansible script, we will modify the <environment>-front.yml file to set configuration of Magento to point static file in S3 bucket created by Terraform.

**Which questions would you ask to know better the current deployment workflow and possible integration with Cycloid ?**

* Could explain more in deep Web App Architecture? Components and their HA characteristic (Actif/Actif, Actif/Passif, Cluster, Farm)?
* Are you using Container or VM?
* Do you have DevOps tools chain in place? Could you describe it?
* What is the LiveCycle of your applications?
* How do you build and deploy your applications?
* Do you have Release pipeline?
* How do provision your infrastructure?
* How do you scale up your infrastructure to handle spikes of traffic?

**If you had to advise the customer on migration on Aws, what would you propose to the customer as HA architecture design?**

**Explain how the proposed components should interact with each other?**

Assuming that the architecture is 3 tiers applications: Webserver web page, Api Service Server, Data Base.

I will suggest building 2 subnets behind a VPC. Each Subnet is located in different region. Front of 2 subnet set 2 Elastic Load Balancer.

ELB 1 manage web server layer in auto-scaling group to ensure scale up/down in case of spikes of traffic and spin up a new instance in case of server fail down. Those servers are connected to App server through ELB 2. To serve Static page, we can use S3 bucket and forward traffic to S3 Url by using Cloudfront who is natively HA.

ELB 2 manage app server like ELB1 in that low layer.

For Database, we will create RDS multi-zone which mean creating 2 instances of Database, set as Master/Slave replicat. Master will serve app server in case of fail over on master, slave will take care of serving app server.

Design below show HA architecture describe above:



**Try to integrate Terraform, Ansible and Packer technologies in your design. How would you use them, for which purpose ?**

In that desing, we will use Terraform to create the cloud infrastructure and set the typologie of server used (Create VPC, subnets, define AMI characteristic, create IAM, set Auto-scaling groups, Create RDS instances)

Ansible will be build, install the servers and configure servers to be connected. We will use packer to snapshot server created by ansible to create a new AMI base that ELB will use in auto-scaling groups to spin up new server.

**Bonus : what would you change if the application is container ready ?**

Instead of creating AMI, we will use Packer to snapshot container images and save images in aws docker registry.