Infra Transformation through complete automation

Customer: One of the top media solutions company

Summary

Powerup cloud helped the customer completely transform their business environment through complete automation. Our design architecture and solution engineering improved business process efficiency, without any manual intervention, resulting in turnaround time being decreased by more than 90%. Now most of their applications running on the cloud, the customer has become one of most customer-friendly media companies in India.

Problem Statement

The customer’s team wants to concentrate on building applications rather than spending time on the infrastructure setup and dependencies packages installed and maintained on the servers. The proposing solution needs to be quick & scalable one so that business performance will be improved significantly.

Proposed Solution

Focusing on workload and transaction volume, we designed a customer-friendly, network optimized, a highly agile and scalable cloud platform that enabled cost optimization, effective management, and easy deployment. This helped reducing interventions and cost overheads.

CloudFormation Templates:

We used AWS native tool CloudFormation to deploy the infrastructure as code, the ideology behind this is deployed infra as well as we can use it for Disaster Recovery.

CloudFormation template implemented in Stage & prod environment based on the best practice of AWS by subjecting the servers to reside in private subnets and internet routing with the help of Nat-gateway.

To remove the IP dependencies for a better way to manage failures, the servers and the websites are pointed to the Application load balancers where a single load
balancer we managed to have multiple target groups in the view of cost optimization.

Base Packages Dependency:

This solution must remove the dependency of the developer to install the packages on the server to support the application.

The packages need to be installed on the infra setup, so the developer can deploy the code using the code deployer services rather than spending time to install dependencies.

Hence, we proposed & implemented the solution via Ansible, With the help of ansible we can able to manage multiple servers under a single roof. We have prepared a shell script which will install the packages on the server.

The architecture majorly differentiated in the means of Backend & frontend Module.
Backend Module where the java application will be running, hence a shell script will run the backend servers which will install the Java-8 versions and creates a Home path which based on standard path, so home path execution of application will be always satisfied by this condition.

Frontend Module which more likely of Nginx combined with node.js which achieved by the same methodology.

**Volume Mount**
The application’s logs and other backup artifacts are managed in the secondary EBS volume which the mount point to the fstab entries are also automated.

**Code-deploy agent**
The Main part of deployment achieved by the code-deployer hence the servers should be installed with code-deployment agents during the server setup which is also done through ansible.

**User Access:**
User access is another solution, where the access to the servers restricted for some people in the development team and the access will be provided to the server with the approval of their leads.

We had, dev, qa, psr, stage and prod environments we clubbed all the servers in the ansible inventory and generated a public key and private key and passed them on the standard part. When the user adds scripts runs, ansible will copy the public keys and create a user on destination server by pasting the public key in the authorized file.

This method will be hidden the pub key from the end-user when the user asked to removed using ansible we will delete those users from the server.

**Monitoring with sensu:**
Infra team is responsible for monitoring the infra, hence we created a shell script which will install the sensu on the destination server for monitoring using ansible.
By implementing these solutions, the development was less worried about the packages dependencies which allowed them to concentrate on their app development and fixing bugs and user access got streamlined.

**Bastion with MFA settings:**

The servers in the environment can get accessed only by bastion server which acts as the entry point.

This bastion server was set up with the MFA mechanism, where each user must access the server with MFA authentication as a security best practice.

**SSL renewal**

In one of the legacy account, SSL offloaded in the server level with a lot of Vhosts. Hence renewing of certificates will take time to reduce the time we used SSL with ansible to rotate the certificates in quick time with less human efforts.

**Automation in Pipeline :**

1) Terraform implementation
2) Base packages installation on bootup which reduces one step of installation.
3) User access with automatic expiry condition.

**Challenges**

In addition to the on-going consulting engagement with the customer for enhancement, and designing a system to meet the client’s need, Powerupcloud also faced some challenges. The Infra has to be created in quick time with 13 servers under the application load balancers, which includes Networking, compute and load balancers with target groups. The Instances were required to install with certain dependencies to run the application smoothly. As a result, the development process became more complicated.

The solution was also expected to meet the very high level of security, continuous monitoring, Non-stop 24X7 operation, High availability, agility, scalability, less turnaround time and high performance, which was a considerable challenge given the high business criticality of the application.
To overcome these challenges, we established a predictive performance model for early problem detection and prevention. Also, started a dedicated performance analysis team with active participation from various client groups.

All the changes in configuration are smoothly and rapidly executed from the viewpoint of minimizing load balance and outage time.

**Business Result & outcome**

With the move to automation, the customer’s turn-around time decreased by 30%. This new system also helped them reduced capital investments as it is completely automated. The solution was designed in-keeping with our approach of security, scalability, agility, and reusability.

- Complete automation
- Successful implementation of CloudFormation template.
- Improved business process efficiency by over 90%
- Network optimized for a virtualized environment.
- Key-based access Mechanism with secured logins.
- Highly agile and Scalable environment.

**Cloud platform**

AWS.

**Technologies used**

Cloudformation template, Ansible.