

Capstone Project Proposal 2018

One Touch (Infrastructure as Code)

JANUARY 31

Athira Sudakaran Pillai – C0696574

Nithin Liviero – C0688727

Sidhant Gupta – C0697252



ABSTRACT

Automation has become the key to efficiency in IT industry. Automation leads to high quality deliverables in lesser time. Project managers also love automation in the projects which helps them in managing the timeline and cost of the projects. But when it comes to IT, the most useful area of automation is in IT infrastructure. System Administrators are the one who try to implement it more for the right reasons. And nowadays Infrastructure as code is considered as a most appropriate process of building infrastructure services rather than doing manually.

Both developers and administrators incorporate infrastructure using different configuration files as software codes. These set of codes help in maintaining different services such as compute, networking, applications and storage collectively as OS. IaC eradicates the uneven outputs as a result of human error, thereby; amplify the speed and agility in infrastructure deployments. IT industry is moving to a phase where in all the infrastructure requirements are automated in some way or the other and this project gains its relevance in this space because of the extra level of automation that we are trying to bring into this whole process. And Amazon Web Services with its wide variety of services provides the perfect platform for us to do this. While considering the capability of different services in AWS we think our goal is ambitious but achievable.

While we aim to achieve a complete one-touch infrastructure deployment solutions of the Cloud administration, the dream is to make it zero touch in future. Through this project we will also try to integrate all services provided by AWS “Alexa” to make the infrastructure deployment more seamless.

While considering AWS and IaC together, administrators gets simple solution to handle different services in a proper and controlled manner. AWS Cloud Formation templates which will be used for this project can be re-used to create similar copies of the stacks across the AWS regions and accounts.

INTRODUCTION

Automated environment deployment is a dream solution for any system administrator. Although we are able to achieve some of those in the traditional on-premise infrastructure, the level to which we can scale it is very limited. But when it comes to doing the same in cloud service offerings the possibilities are

endless. Taking that into consideration we are trying to make an automated infrastructure deployment solution in AWS. We have decided to make use of the CloudFormation services working together with other AWS services like Lambda, Cloudwatch, Lex etc to achieve this goal.

AWS CloudFormation is a tool that provides developers and organisations a simple path to build group of different AWS resources and execute them in a good manner and also in accordance to what trends are going on. AWS Cloud formation is responsible for all the resources to implement in a proper way and also focused on the dependencies between them and to execute an application, AWS assets are required in the form of simple text file called template.

AWS Cloud formation let you create and re-create applications and infrastructure and also perform right functions to organise your stack, diagnose errors and debugging automatically to avoid or reduce manual actions and also permit resources to be secure and reusable.[2]

The proposed project is aiming to use most of AWS services which include computing resources like EC2, Storage services S3 bucket, Monitoring and logging tools such as CloudWatch, networking and content delivery services such as VPC,API gateway,CloudFront and data base services like Relational Database Services and other associated resources.

Lambda is one of the most popular services in AWS, is a critical component of our project. Lambda supports multiple scripting languages such as JSON, Python, node.js, Java etc.. For our project, we intend to use Python 3.6 for automatically configuring the infrastructure. One of the striking features of AWS Lambda is that it lets you run code without managing or provisioning servers. It integrates all types of application or backend service with minimal administration. As a part of our project goals, we will implement an Infrastructure environment with all supporting services using the code already deployed into Lambda, with a single click (or voice command from Amazon Alexa). With each trigger, Lambda can scale the application by running the code as a response. CloudWatch helps with scheduling the Lambda triggers for a particular time instance. We would be using CloudFormation functionality once we have the Python script and Lambda environment ready. AWS CloudFormation enables us to deploy infrastructure resources and applications by accessing the resource specifications from the user specified YAML files.

Once the Lambda and CloudFormation environments are established, we are planning to deploy a Virtual Private Cloud (VPC) for multiple subnets with NAT enabled where required. Route 53 and API gateways are also a part of the networking aspect of the project. Multiple EC2 instances would be deployed within these VPC subnets with connectivity to the backend databases like RDS and storage S3 buckets.

All these work together with AWS Identity and Access Management (IAM) allows you to give different users or group of users of your choice can access or deny specific services and resources safely and securely with multi factor authentication.

PURPOSE

The main use case for this project is to provide a new way for the cloud administrators to build multiple infrastructures of same configuration with minimal effort and human interaction as possible. Usually in an industry environment there will be a need to build multiple infrastructure with the same configuration for a single project which happens in different stages like development testing, Pre-production, production etc. Our solution will be very useful in these cases.

STATEMENT OF NEED

The proposed project we will be developing IaC as code in CloudFormation to spin up infrastructure with same configuration. Cloud administrators are always looking for ways to automate their jobs to spend their time more effectively to learn and implement new things in their infrastructure. So any kind of automation of the current process is highly useful for them. Time spend for setting up the automation process would prove really fruitful in future and it is a proven fact. There is a need of the solution that we are proposing because while it help to save time to create to environments, it will also help to reduce the human errors .

LITERATURE REVIEW

An extensive literature review was carried out from various sources such as IEEE, e-books, websites and video tutorials from Coursera and Udemy. Also, we had contacted the experts who work in the AWS cloud environment to gather additional information and get a real world perspective.

Infrastructure as a Code (IaC) is one of the basic requirements for DevOps practice and finds great value in functions like version control, automation and continuous integration. According to Puppet.com, who

is one of the pioneers in the field of Cloud automation and DevOps, Infrastructure as Code is a software that creates an automated environment which is capable of managing and provisioning on its own. Amazon AWS' s cloud formation provides a common language platform to provision infrastructure resources in a virtualized environment. CloudFormation integrated with Lambda has a great potential in automating AWS cloud environment and is repeatable, testable and less susceptible to errors. To AWS team this single file of code serves as a “source of truth for cloud environment” [2]. The white paper published by Amazon Web services on July 2017 which was titled “Infrastructure as a code” gives an in-depth knowledge as to how AWS treats IaC in their cloud environment[3]. An IEEE paper titled “DevOps: Introducing Infrastructure-as-Code” gives us a clear idea on how DevOps methodology has efficiently incorporated IaC in cloud orchestration [4]. The video tutorials from Coursera about cloud infrastructure monitoring and automation using various tools like Jenkins, Ansible, Terraform was an eye opener for us and gave us enough confidence to go about it with an innovative way to utilize IaC in the AWS cloud infrastructure [5]. The AWS Lambda is a server-less compute service that runs the code in response to events such as HTTP requests and automatically provisions and manages the underlying cloud resources. It can even invoke the backend resources [6]. Another factor which makes automation in AWS environment interesting is the Amazon CloudWatch Events which is a near real-time system in response to events occurring[7].

METHODOLOGY

We have decided to follow scrum methodology for this project.

- We will be setting up different sprints to achieve each part of the project.
- We shall also have daily call to evaluate the progress and identify the pain point (instead of having a daily stand ups).
- All the developed scripts will be tested thoroughly and integration will be done by the end of each sprint.
- Any reconfiguration to the design will be done if required as the project progresses.
- We will also have a bi-weekly team meeting in which we will be discuss and setting the goals for the sprints for the coming two weeks.
- All the members in the team will be responsible to complete their part on time and test the scripts.

- One person in the team will be responsible to double check the integration and make sure things are working fine.

EXPECTED RESULTS

Once this solution is complete we expect it to be a fully automated scalable and redundant solution for spinning up multiple infrastructure environments. The resources that are used to build this solution will be highly reusable. These resources can be customized with minimum effort and will be easy to configure and implement in a different project.

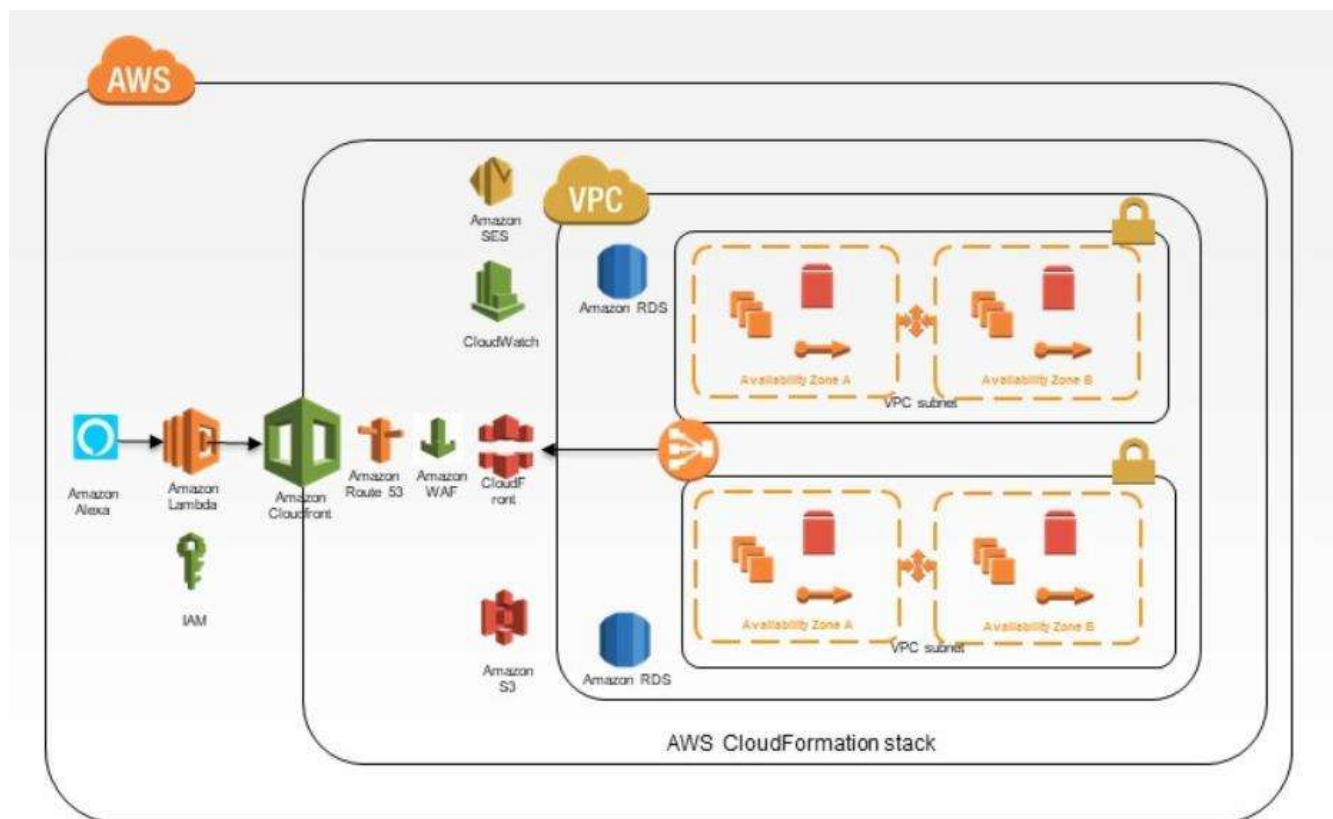


Fig: Architecture Diagram*

TIMELINE

The timeline is breakdown to seven tasks starting from project discussion which we had started from January 5th to finalize the topic and necessary literature to substantiate the feasibility of the proposed project. WE have started to develop the python code for CloudFormation as task 2 and expected to reach task by end of February. Task 3 is the extensive code testing to fix any possible bug present and as a part of Task 4 we will be integrating the AWS services with the tested code. Task 5 includes a vigorous testing

that covers the connectivity test as well as the infrastructure components functional validation. Task 6, which will be commencing from beginning of April includes the implementation of the complete infrastructure as specified in the proposal above.

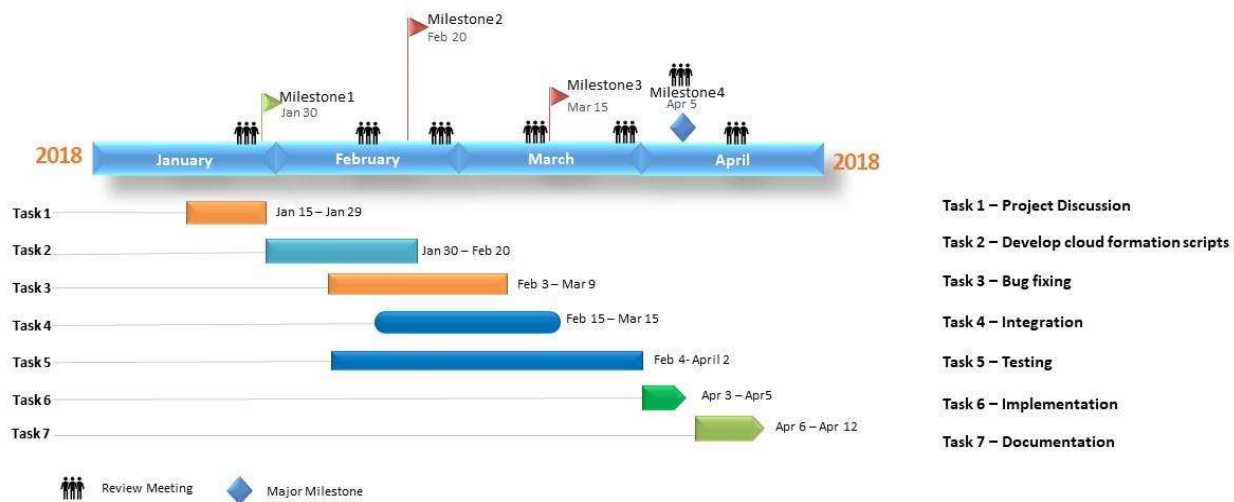


Fig: Timeline

*Diagram shown in the report is not the exact infrastructure we will be building. There might be few changes based on different factors.

REFERENCES

1. <https://puppet.com/products/why-puppet>
2. <https://aws.amazon.com/cloudformation/>
3. <https://d0.awsstatic.com/whitepapers/DevOps/infrastructure-as-code.pdf>
4. <http://ieeexplore.ieee.org/document/7965401/>
5. <https://www.coursera.org/specializations/cloud-computing>
6. <https://aws.amazon.com/lambda/features/>
7. <https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/WhatIsCloudWatchEvents.html>
8. <https://cloudacademy.com/blog/aws-security-identity-and-access-management-iam/>