

# **Migrating Existing ECS Application to Kubernetes**

---

# TABLE OF CONTENTS

|                              |   |
|------------------------------|---|
| 1 About Our Client           | 1 |
| 2 The Need to Migrate        | 2 |
| 3 A deep dive into Terraform | 3 |
| 4 Rancher                    | 4 |
| 5 How our team at ACC helped | 4 |
| 6 The need for load balancer | 5 |

## Our client a leading Entertainment company

Our clients never compromise on technology and for them technology is the backbone of the content creation ecosystem, supporting features like seamless content streaming, highly advanced data analytics, a powerful ad suite, and first-rate asset management. enabling them to interact and engage with the audience more effectively.

Achieving 1.3 billion viewers globally across linear and digital platforms after launching India's first private satellite TV station in 1992. Today, Our client is the go-to source for entertainment throughout the world thanks to an integrated team that produces and provides exceptional content. It takes a lot of skill to provide customers with entertainment around-the-clock in a variety of languages across a vast array of geographical locations. Their activities are centered on cutting-edge technological foundations, which also fuels their capacity for innovation.



## The need to migrate to Kubernetes

Kubernetes is a distributed system resiliency platform. It manages the scaling and failover of your application, as well as deployment strategies. For example, Kubernetes can easily manage your system's canary deployment. Kubernetes can be used for load balancing and service discovery.

Kubernetes restarts failing containers, replaces them, destroys containers that don't respond to a user-defined health check, and doesn't broadcast them to clients until they're ready to serve. It helps organize your storage, Kubernetes lets you mount any storage system you choose, including local storage, public cloud providers, and more.

Kubernetes enhances the development, release, and deployment processes much smoother: for example, it enables container integration and makes managing access to storage resources from many providers much easier.

Furthermore, with microservices-based architectures, the application is divided into functional units that communicate with one another via APIs, allowing the development team to be divided into smaller groups, each specialized in a certain feature. This structure enables IT teams to work with more focus and efficiency, resulting in faster release times.

## KUBERNETES' ADVANTAGES FOR BUSINESSES

- Upgrades and deployments can be controlled and automated.
- Save money by getting the most out of your infrastructure resources by making better use of your gear.
- Container orchestration on a large number of hosts
- Many of the problems produced by container growth can be handled by grouping them into "pods."
- Scaling of resources and applications in real time
- Autocorrection and application testing

## A deep dive into Terraform

Terraform is an open-source tool that utilizes declarative configuration files to provision Cloud services including virtual machines, containers, storage, and networking. Terraform's infrastructure-as-code (IAC) approach helps you follow DevOps best practices for change management by allowing you to manage Terraform configuration files in source control to keep your testing and production environments in top shape. Through its application programming interfaces, Terraform develops and manages resources on cloud platforms and other services (APIs). Terraform can work with almost any platform or service that has an API that can be accessed using providers. Terraform users employ HCL, a JSON-like configuration language, to specify and enforce infrastructure configurations. HCL's straightforward syntax enables provisioning and re-provisioning infrastructure across different cloud and on-premises data centers simple for DevOps teams.

With self-service infrastructure as code and integrations with VCS, ITSM, and CI/CD, the infrastructure automation workflow can be extended to all teams in the business.

It basically means to install Terraform on your machine and write Terraform manifests. When you run the `terraform apply` command, it creates the infrastructure in the cloud according to your manifest. Since Terraform is mainly used to create new resources, its manifest files define, for example, the type of virtual machines and how many of them should be deployed. After Terraform is done provisioning, it saves the details of the provisioned resources (for example, IP addresses of created virtual machines) back into the state file.

## Rancher

Most developers don't care about the technical aspects of IT infrastructure. They want to write code securely and use their preferred development workflow. Rancher simplifies Kubernetes-as-a-Service deployment for development teams with push-button deployment of their containerized workloads on-premises, in the cloud, in hybrid deployments, or at the edge. Meanwhile, operators can retain full administrative control and observability.

Rancher makes it simple to handle security across clusters. You can create users, even using external authentication techniques like LDAP, assign permissions to these users, and assign them to resources in any Rancher-managed cluster.

Rancher provides all of Kubernetes' functionality in a user-friendly console. You can use containers to develop projects, manage workloads, define replicas, load balancer setups, and other things.

## How our team at ACC helped

The client had to migrate the existing application to kubernetes. The application already existed on docker which was migrated to kubernetes for continuous integrations the team had access to the backend only as the existing process was on docker transferred to kubernetes all the infrastructure was developed on terraform by using all the latest tools which the team had to research upon for all the latest possible solutions and then the infrastructure code was written and rancher was used for monitoring the application deployed on kubernetes.

As only those who dare to fail greatly can achieve greatly, we had a dedicated team to research on all the latest tools available to make the application efficient, the preferred load balancer was the application load balancer over the network load balancer the load testing work is still in progress.

## The need for load balancer

The Application Load Balancer is a feature of Elastic Load Balancing that allows a developer to configure and route incoming end-user traffic to applications based in the AWS public cloud.

The OSI model has a seventh layer, the application layer, is where application load balancers operate. Application load balancers are designed for cloud environments, to evaluate application content and route it to applications on the AWS public cloud. In the beginning, it analyzes packet details and application-level contents rather than port and IP, such as HTTP, HTTPS, and packet details. Then, based on the incoming HTTP-formatted traffic, it divides the traffic to the target group according to availability. Similar to other load balancers, application load balancers make sure that no server is overloaded by the volume of traffic. The application load balancer may do health checks on your servers, which makes it special. It helps to find out possible coding and HTTP errors.

The importance of load balancing cannot be overstated in a cloud environment with several online applications. A load balancer makes sure that no single server is subjected to an excessive amount of demand by dividing network traffic and information flows among several servers. In addition to enhancing user experiences and protecting against distributed denial-of-service (DDoS) attacks, this increases application responsiveness and availability.

This is the distribution of requests from the network layer to the application layer based on various criteria. It is aware of context and can quickly guide requests based on a single variable or a group of variables. Instead of relying exclusively on server (operating system or virtualization layer) information, applications are load balanced based on their particular behavior.

In comparison to the Classic Load Balancer, Application Load Balancer is more flexible and offers additional functionality in addition to cost savings.



Application Load Balancer also supports the following additional cutting-edge features:

- Using content-based routing, it is possible to create applications with numerous microservices that may be scaled independently in response to the demand placed on various services
- Target groups, which can exist independently and are logical groupings of servers behind a load balancer
- Apps that use containers, where a single instance can host a number of containers that listen on various ports for network traffic coming from the same target group
- Carrying out fine-grained port-level health checks; and Access logs that contain additional information and are saved in a compressed format.





# ABOUT ACC

ACC is an AWS Advance Partner with AWS Mobility Competency. Awarded The Best BFSI industry Consulting Partner for the year 2019, ACC has had several successful cloud migration and application development projects to its credit.

Our business offerings include Digitalisation, Cloud Services, Product Engineering, Big Data & Analytics and Cloud Security. ACC has developed several products to its credit. These include Ottohm – Enterprise Video and OTT Platform, Atlas API – API Management and Development Platform, Atlas CLM – Cloud Life Cycle Management, Atlas HCM – HR Digital Onboarding and Employee Management, Atlas ITSM – Vendor Onboarding and Service Management and Smart Contracts – Contract Automation and Management.



[www.appliedcloudcomputing.com](http://www.appliedcloudcomputing.com)

**Shubho Pramanik**

**(+91) 90297 20294**

**shubho@acc.ltd**

**Rogin Rappai**

**(+91) 8828478321**

**rogin.rappai@acc.ltd**

also reach us at

