



About our client

Blacksheep OTT started their Journey on YouTube as a 'Smile Settai' channel in 2013 with a team of 20 people. They are unique in delivering all content which makes them reach 8 Lakhs in a Span of 4 years. Later, they restarted on YouTube as a 'Blacksheep' channel in 2017 with a team of 40 people. They reached 3 million followers in a short span of 2 years. Now, they are with 3.5 million subscribers in Blacksheep with the team size of 200+ Also, with one more channel named 'Unakkennapaa' in the year 2019 which is around 2 million subscribers now.

As an extent, they started 4 Non - Fiction Channels in 2020 named Blacksheep Cinemas, Blacksheep GO, Blacksheep Darbar and Blacksheep Sports which is around 1 Lakh subscribers in 6 months. Finally, they started their own OTT Channel named 'BS Value' in Nov 2020 which is now with 5 Lakh downloads in 3 months.

Customer Challenges

1. Current Infrastructure built was inelastic, unreliable and not fault-tolerant, due to which the production environment often used to crash due to traffic spikes during any new web series launch.
2. Infrastructures were provisioned manually so the drifts and changes were not maintained or not versioned which caused lots of human errors in maintaining infrastructure across environments.
3. Slow provisioning of Infrastructure if any new dev/test environment was needed to test new features so release new feature was taking months.
4. Automated Build, Deployment and Rollback was required to reduce manual intervention by developers in production. The changes to both the application and infrastructure were desired to be automated.
5. Monitoring and logging of Application and Infrastructures was missing.

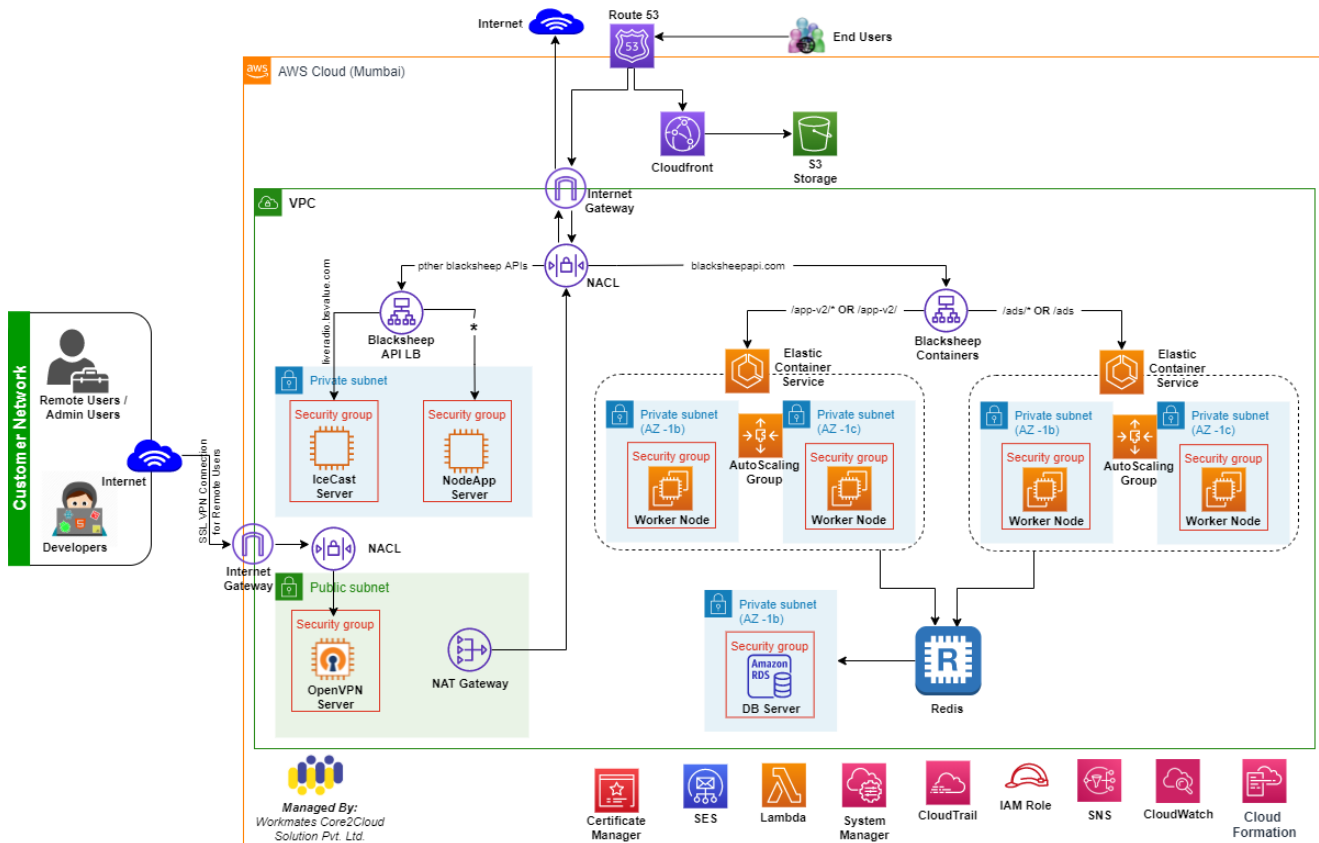
Solution Approach

1. We based our solution approach by leveraging managed services in AWS, such as VPC, RDS, EC2 and CloudFormation. We used AWS cloud Formation templates to automate the process of provisioning infrastructure on AWS by developing templates using YAML.
2. To validate the syntax of the templates we used CFN-lint and CFN to verify the structure of the YAML template before deploying it into the AWS account.
3. For BSValue, we defined parameters within the template and ensured that **no explicit naming** was used for creating any custom resources.
4. We achieved this by defining friendly parameters within our template for each AWS service needed to be deployed. Eg. Instead of specifying parameters with naming conventions. we used parameter groups to define namespaces for just the category and capacity of the services meant

to be provisioned.

5. Also included in the template was an outputs section which allowed us to display the outcome of the stacks deployed by the template.
6. The outputs section of the stack further allowed us to map the values onto AWS Systems Manager Parameter Store to avoid exposing server credentials such as SSH keys and RDP Password.
7. In addition to the parameter groups and stack outputs, BSValue also had a use case to implement AMI based deployments. For this we used source mappings for auto fetching region and AMI ID values.
8. BSValue's infrastructure involved deploying multiple resources within a single VPC. We solved this by using nested stacks in our template by defining Pseudo parameters for each of the correlated resources.
9. Before each segment of the stack executed, our template had logical syntaxes to fetch the metadata of each parameter and the output before moving on to the creation of the next resource within the stack.
10. The Architecture of BSValue leveraged application modernization by means of Elastic Container Service.
11. We helped BSValue by defining a CFN template that had the option to reference metadata for information such as Region (AWS::Region) and specific parameter types to define the credentials for accessing the EC2 instances within the cluster (AWS::EC2::KeyPair::KeyName for ssh keys).
12. Additionally, our devops team focused on creating a template that had the brevity to define pseudo parameter for each of the auxiliary resources such as IAM Role, IAM user and Instance Type within CFN.
13. For automating background jobs, backup and DR activities, we leveraged AWS SAM (Serverless Application Model) to define specific lambda functions for infrastructure backups and certain background jobs that were necessary for the Microservices deployed within the ECS cluster.
14. Our team made the best use of cfn-pylint and cfn-guard to validate and rectify the templates for any security exposures, syntax issues or loopholes. Therefore, entrusting BSValue with an example of DevSecOps.
15. We have also leveraged AWS Code pipeline and code deploy to manage deployment patterns for future build or releases. Code pipeline enabled us to orchestrate the flow of deployment from development to production. Whereas, Code Deploy allowed us to implement various methodologies for implementing build and release patterns.
16. We used In-Place rolling updates deployment pattern within CodePipeline for future builds / releases.

Solution Architecture



Key Services Used

VPC, CloudFront, CloudTrail, CloudWatch, EC2, SSL VPN server, Cloud trail, Cloud watch, Lambda, Auto-Recovery, SNS , ECS etc.

Results and Benefits

1. The level of automation through the IaC deployment allowed faster time to market
2. Our IaC approach allowed townhall to avoid 'configuration drifts' tends to cause mismatched development, test, and/or deployment environments due to ad-hoc configuration changes and updates. Thereby, enabling a more efficient development and production lifecycle.
3. Our approach towards DevOps methods helped BSValue implement CI/CD which ensured zero downtime and quality assurance.
4. Albeit initial costs were provocative but in pre course of the devops implementation itself BSValue witnessed a significant reduction in overall infrastructure costs and increased TCO for managing the infra for their application code.
5. A mutable infrastructure avoided the calamities caused by human errors , as IaC model allowed clear visibility of the changes occurred and the option to reverse the effects as well.
6. BSValue overcame the problems of slow provisioning by a leveraging microservices architecture with more agile deployment patterns for shipping new releases.



About Workmates

Workmates core2cloud Solution Pvt. Ltd is an AWS Advance consulting partner and Leading Cloud Management Company in Eastern India. Workmates Core2cloud is a cloud managed services company focused on AWS services, the fastest growing AWS Advanced Consulting Partner in India. We focus on Managed services, Cloud Migration and Implementation of various value-added services on the cloud including but not limited to Cyber Security and Analytics. Our skills cut across various workloads like Microsoft, SAP, Media Solutions, E-commerce, Analytics, IOT, Machine Learning, VR, AR etc. Our VR services are transforming many businesses.

Workmates has a yellow theme which is the color of youth. Our Vibrant team of 100% certified resources brings the edge to customers for an End to End AWS partner, committed towards quality and supporting their business on a 24X7 basis.