DevOps Implementation for
Ruah Technologies and Solutions

About Customer: Drsignet Healthcare

It is a platform developed by DrSignet to provide an all round ecosystem for providing smarter healthcare through Technology. It will act as a hub to bring different stake holders of the healthcare chain together.

Imagine waiting in line to get a doctor's appointment, a test centre or at a pharmacy store to get medicines, you have to spend a lot of time. getting through formalities while taking care of your ailing loved one. Well, this is what healthcare has come to these Days.

Now meet DrSignet, a platform that can provide you all the medical services right from the comfort of your home. You can book doctors, tests and order medicines just with your smartphone. No long queues anymore!

Customer Challenge

Drsignet was building the 2 mobile applications to run on the AWS cloud. As the application is in development phase and customer required the secure and reliable application configuration on AWS cloud. DrSignet was looking for expert guidance for implementing the following key requirements.

- To configure the multiple mobile applications with Nginx reverse proxy configuration on top of the private EC2 instance.
- Leaset privileges for application developers to push and pull only. Also, want to enable the deletion protection of code repositories for each user.
- NodeJS application logs were stopped unexpected intermittently.
- Required the database credentials encryption and don't wanted to store credentials in code repository.
- To create a cloud native deployment architecture for the OTT Platform and web application with the best practices in place.
- Regular Mongo Backup mechanism required in case of server failure.
- Monitoring of the AWS production resources was the main requirement to troubleshoot and debug the infrastructure issues.
- Rollback mechanism was not there for the production resources. And alert management was required during the new deployment.

DrSignet collabes with Workmates to address out the above requirements. Drsignet wants the fixes of the Nginx reverse proxy permanently on the AWS Cloud. Also, make it DevOps compliant with secure, high-performing, resilient, and efficient infrastructure for their both mobile applications.

Our DevOps Solution Approach

Workmates, an AWS DevOps specialist, helped DrSignet adopt the DevOps methodology to their software development and release processes. The application architecture for the AWS cloud was designed with the right balance of managed services to both reduce operational overhead while keeping the costs low. Most of the undifferentiated activities on the cloud was automated, thus reducing the overall time and risk in deployment of new services. The key aspects of the solution design includes:

- Templates for provisioning the collection of resources together as a single unit.
- 2 tier-based architecture for the react web application & Mongo server. Also, APIs are residing in react server itself.
- Amazon CloudWatch Events to automatically start the configured AWS pipeline when a change/release occurs in AWS Code Commit repository.
- A fully featured and automated CI-CD pipeline using AWS Code Pipeline for the frontend application and Backend application. The CI-CD pipeline also included the feature for rollback mechanism when the deployment fails and failed notification alert will be triggered to customer.
- Application load balancer endpoint is used for updating the Doctor/patient records, appointments, pushing the notification to doctor/patient from the Web Admin Portal.
- Mongo backups are stored on S3 storage service with 15 days retention and can be accessible and restorable at any point of time in case of server failure. It was scheduled using shell script and it run every midnight 12 pm sharp. RTO is max 2 hours and RPO is 24 hours, which can be achieved to perform disaster recovery.
- In-place deployment type is adopted and using the OneAtATime deployment strategy which is updating the instances in the AWS deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline from the application load balancer for its update.
- S3 Storage in S3 buckets and
- Configured the tags for EC2 instances and have registered in load balancer to manage incoming traffic during the deployment process. The load balancer blocks traffic from each instance while it’s being deployed to and allows traffic to it again after the deployment succeeds.
- Appspec file is placed in the root of the repository which enables the developer to push the code changes in production environment.

- Infrastructure monitoring enabled using AWS CloudWatch service and application logs can be achieved from the server itself using the opensource pm2 tool.
- AWS Config setup for Continuous Monitoring, assessment and change management for the AWS resource's configurations.
- IAM role attached to the EC2 Instances for uploading the Mongo Backup. Specific S3 event policy (Inline Policy) is assigned to the Role for performing the backup mechanism.
- SSM Patch Manager configured to scan EC2 instances and report compliance on a schedule, install available patches on a schedule, and patch or scan instances on quarterly basis.
- AWS Code Commit repositories having only source code and database secrets and credentials were stored in S3 storge bucket. The secrets get offloaded during deploy stage on the CICD pipeline for both applications.

AWS Services Used

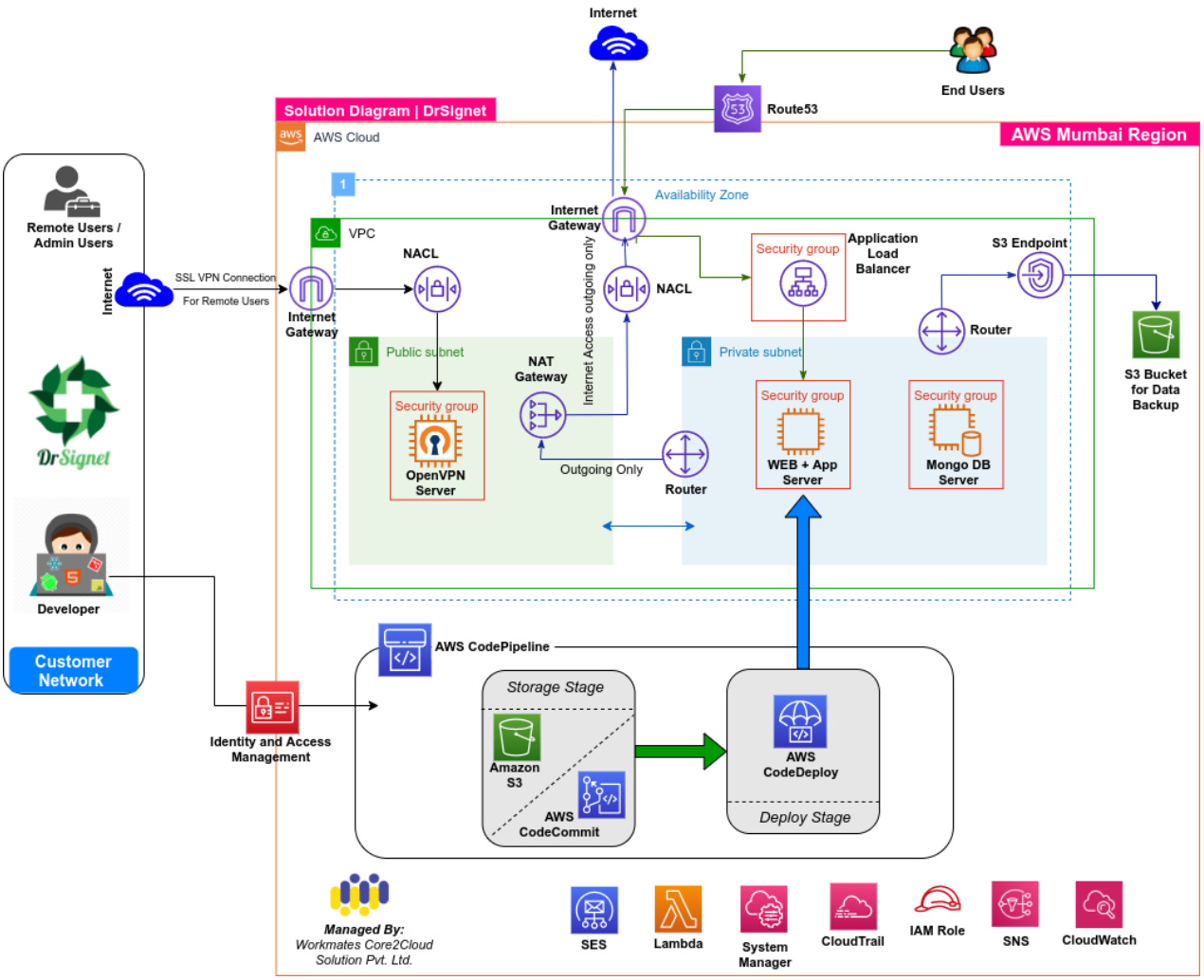
AWS Services Used	Use Case
AWS EC2	For hosting the Mongo database and Angular web application
AWS Application Load balancer	Public endpoint for accessing backend Admin portal only.
S3	Object Storage for database backup and logs
AWS KMS	For EBS and S3 encryption
AWS Code Commit	Codebase repository for frontend and backend application
AWS CloudFormation	Creation of isolated VPC, OpenVPN server
Amazon CloudWatch Logs	Logging Solution for all microservice applications
AWS Config	Conduct assessment and audit of the AWS resources
AWS Systems Manager	For On demand Patching EC2 Servers.
AWS Code Deploy	For deploying the new releases using in-place deployment type.
AWS Code Pipeline	Creating pipeline for deploying the application code from source to AWS Prod environment.
AWS Resource Group	For creating group to enable the patch cycle through SSM

Third-Party Tools	UseCase
OpenVpn	Secure access for AWS resources

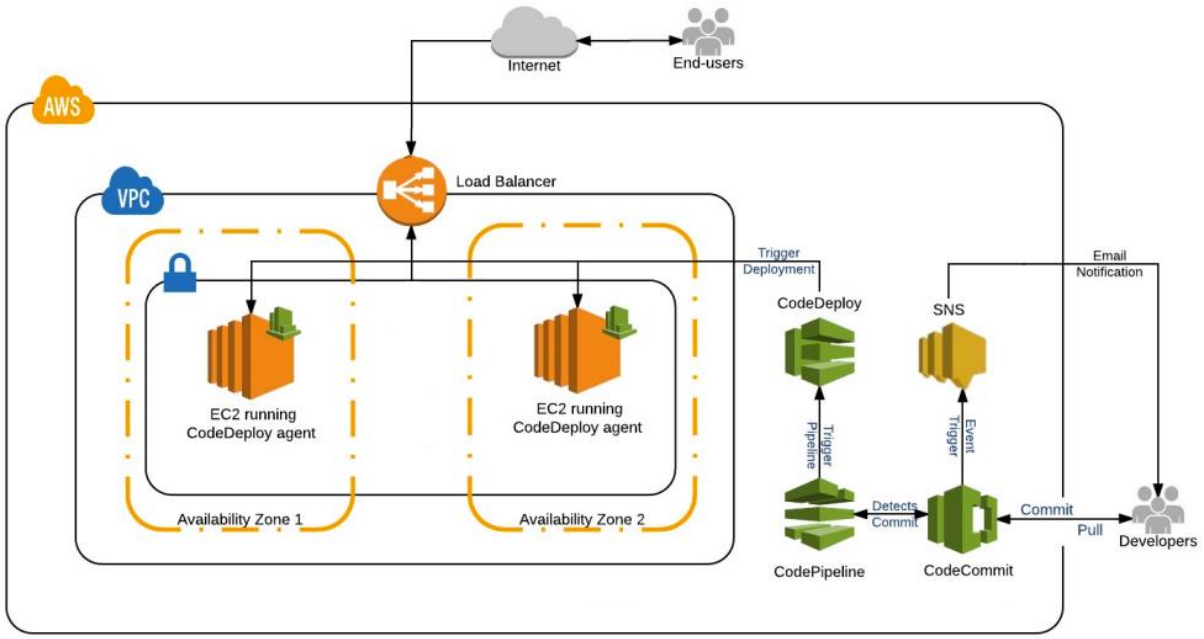
Security Considerations

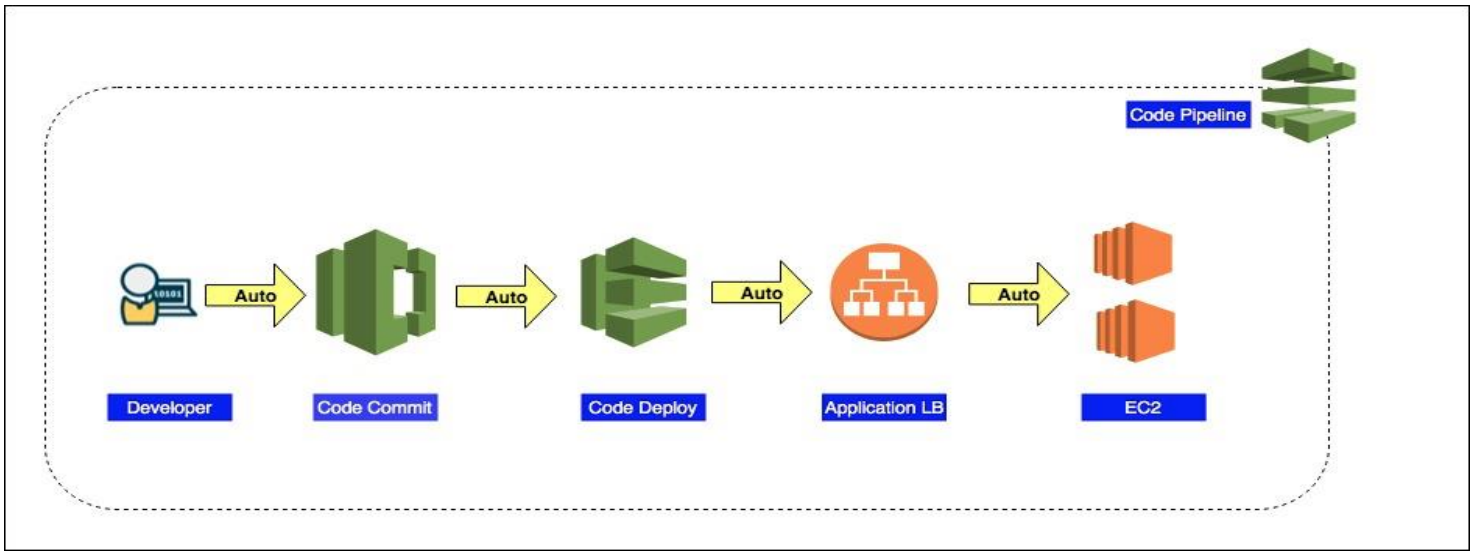
- AWS Asynchronous KMS Encryption is enabled for all provisioned S3 Storage buckets and in transit data is encrypted using AWS ACM.
- AWS IAM role-based access control to restrict users to the required resources only. Custom and Inline policy are attached to AWS IAM Roles. Specific events actions are place into the Inline policy.
- Web and Database Server are hosted in private subnet and internet is accessible through NAT gateway for updating the Linux packages and for patching activities.
- Web server is exposed via AWS application load balancer and both applications are serving traffic through different ports.
- DB Admins are accessing thier private Mongo databases though OpenVPN only. Mongo database service port is restricted to OpenVPN client and application server only.
- Custom ports implemented for each application and database service. Custom SSH port is used for administrative task.
- Strong IAM password provided to each user. IAM users are given minimal access privileges to AWS resources that still allows them to fulfil their job responsibilities
- Mutli Factor Authentication is enabled for extra layer of protection for Root user name and password.
- CloudTrail logs enabled for tracking all kind of activities performed for AWS resources across all regions. Administrator can get the deep insights of API call of each AWS user.
- AWS Code Commit git credentials are shared with customer and were rotated on quarterly basis.
- Developer can push the modified code to the staging branch and respective Manager approves the changes and merge the changes/releases to production branch. Developers have specific branch policy for pushing the code to staging branch only.
- Frontend and Backend application are hosted in private subnet and deployed on top the Nginx web server.
- S3 Buckets are not in public and can be accessible within the AWS resources.

Solution Architecture



Continuous Integration and Continuous Deployment





The releases/changes of frontend and backend application is triggered once changes merged to production branch on AWS Code Commit. The CICD workflow is configured in AWS Pipeline. Pipeline consist of source (AWS Code Commit) and deploy (AWS Code Deploy) services only. Pipeline containing series of stages during the application feature releases. AWS Code Deploy provides the set of events which are performed at the backend of Pipeline and events are available in deployment history of code deploy console. The CICD stages of pipeline includes the following:

1. Developer commits the changes in staging branch of the repository and informs to the Project Manager.
2. The Project Manager reviews the changes and merges the changes into the Code Commit repository's master branch, Once successfully done, it gets triggered through CloudWatch events and start executing the AWS Pipeline stages. Git checkout is initial stage of the AWS Pipeline.

3. Since frontend and backend applications are React and NodeJS apps, there is no need to create build of the code. So, the next step is running Code Deploy.
4. Now the CD process starts, AWS Code Deploy uses the CodeDeployDefault.OneAtATime deployment configuration which deploys the application revision/releases to only one instance at a time. Deployment process follows below events sequentially.

   BeforeBlockTraffic--->BlockTraffic--->AfterBlockTraffic--->ApplicationStop--->DownloadBundle---
   >BeforeInstall--->Install--->AfterInstall--->ApplicationStart--->ValidateService--->BeforeAllowTraffic---
   >AllowTraffic--->AfterAllowTraffic
5. All above events are configured in deployment groups and included in appsepec.yml. First, load balancer deregisters the single instance from the target group and then we download the .env file from the encrypted AWS S3 storage bucket to the document root of the web server, and then master branch code is deploy into document root of Nginx web server. The status of the deployment stages is also verified from the deployment history for that single instance.
6. Once the deployment is successful for single instance, it starts the above steps for another instance.
7. At last, releases/changes are now deployed in to the private EC2 instance and the status of the deployment and application is verified from the deployment history.

In the above AWS Pipeline, we have also included rollback mechanism, also the status of the deployment failed/succeeded is sent to registered email of the developer team. The Frontend and Backend Pipeline has automatic Rollback feature which is defined in AWS Code deploy. Service. If the Production deployment fails, the pipelines restore the code from the previous deployment ID.

Results and Benefits

Frontend React application and backend NodeJS application was successfully deployed on AWS environment and meets all security & DevOps practices guidelines as per the AWS best practices. The following are some of the key benefits for the customer

1. Continuous integration and Continuous delivery help a lot to the developer to deliver updates quickly and frequently without any downtime.
2. Satisfactory feedback after completing the Nginx and load balancer configuration for multiple applications.
3. After the CI/CD implemented, developers are more focused to build more business functionality in thier application and freed the developers of Infrastructure administration tasks.
4. The DevOps culture enabled the developers and operations teams to achieve their results faster.
5. A secure, reliable and fault tolerant application architecture on the AWS cloud.

6. CloudWatch monitoring and alert actions to notify the Developer and Operations team on any production issues, so they can take action and can mitigate it on immediate basis.
7. AWS native security features are highly secured and data/secrets were encrypted using Asynchronous customer managed key (CMK) and remediating the noncompliant AWS resources using AWS Config service.
8. Lead Time for Changes is very fast and efficient. Also reduces the time, cost, human effort, Maintenace time
9. Faster MTTR (Mean Time to Repair) using automated rollback.

DevOps culture and practices make the developer and Ops team to more focus on their core area of expertise. By Building out the CI/CD pipeline is a success of resiliency, persistence, and drive. Also, DrSignet team has adopted the DevOps practices and they are very much appreciating for this success implementation.

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, DevOps ,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.