Hi, I am going to build a 3-tier architecture in AWS and deploy an employee login java based application. Please do not go with beautification of the application, as the main idea here is to deploy the application in AWS and to see what services we need to configure to make our application and architecture prone to Resilience, Performance, Cost Optimized, Provide better Security and Operational Excellence.

The main idea of this project is to identify the issues at each level and try to resolve them by using the 5 principles:

1. What is the Problem?
2. How to Solve the Problem?
3. How the tool/concept works?
4. Understand the Architecture of the tool.
5. Understand the Installation and Configuration.

Now, the first question?

Why I am deploying my application in Cloud.

Every application has varying load. To support the load, we need infrastructure. The infrastructure in on-premise does not easily support scaling and high availability for varying load. Whenever there is a case of peak load, the infrastructure could hang up and there will be loss to business. If I setup infrastructure to support high peak load, at normal times infrastructure would be sitting ideal. This is also a loss to business. Also it is difficult to predict the peak times.

To solve the above issues I need cloud.

Cloud is like provisioning on demand infrastructure resources by availing the infrastructure through rent from any cloud service providers. This is pay as you go service and most importantly, I don't need to plan ahead like in on premise, I just need to react to the demand which is coming in.

The other reason to opt for cloud is to make use of managed services each vendor offer. I can avoid undifferentiated heavy lifting.

Now, the second question?

Why AWS cloud offering?

1. AWS is leading cloud service provider.
2. Provides many services (200+).
3. Reliable, secure and cost effective.
4. Data Centers all over the world.
5. Has 24 Regions and 77 Availability zones.

Regions: Physical location, where data centers are clustered.

Availability Zone: 1 or more discrete data centers.

By deploying the application in as many regions, make my application highly available and have low latency.

Also cloud offerings has been very useful to start-ups. To setup the own data centers in as many places will involve lot of expenses and it will be difficult to handle by start-ups.

Now coming to AWS project, I am deploying an application in 3-tier architecture that addresses Security, High Availability, Scalability and Resource Optimization.

I should follow certain best practices while deploying application into the cloud.

1. Plan: How we are deploying infrastructure in AWS.
2. Design: Create an Architect diagram to show what are resources/services are deploying and the logical connection between the resources.
3. Deploy: Once the Architecture is ready, I need to create infrastructure in AWS Cloud.
4. Manage & Support: I need to have a plan to manage & support the AWS infrastructure.

Apologies for poor Architecture diagram.

* In the architecture diagram, the instances are deployed as proxy layer, application layer and database layer.
* The end-user first connects Application load balancer URL through Route 53.
* This ALB is configured in public subnet, so that users can connect to the URL.
* The request from ALB is forwarded to web proxy layer and then to ALB.
* The second ALB is configured in private subnet, which accepts request from only proxy web servers.
* The second ALB request goes to the instances in Application subnet and then the request is forwarded to Database instance in Database subnet.
* Instances in both Web proxy layer and application layer are configured with Auto scaling group.
* As all the instances re deployed into private network due to the security reasons, if any administrator wanted to access the ec2 instances to perform the administrator related activities a bastion host is configured.
* It is not a best practice to keep configuration related data in ec2 instances as auto scaling group is configured. I am storing the configuration related data in S3 buckets.
* In order to have the uniform latency across the geographical locations, I have configured cloud front distribution enabled for S3 buckets.

AWS Services/Resources Used

1. Networking

VPC

Internet Gateway

Security Group

Public Subnet

Private Subnet

NAT Gateway

1. Security

Certification Manager

Encryption during transit (SSL Certificates)

Key Management Service

Encryption at rest – KMS – Master Key

1. High Availability

EC2

Load Balancer

1. Scalability

EC2

Auto Scaling

1. Domain Management

Route 53

Route 53 Zone

1. Storage Solutions

RDS

S3

1. Management

IAM

Systems Manager

Bastion Host

Session Manager

1. Monitoring & Alerts

Cloud Watch – Alarms

SNS – SNS Topic

Application Configuration/Deployment

1. Application: Java based employee login app.
2. Application Server: Apache Tomcat 9x
3. Web/Proxy Server: Nginx
4. Database: MySQL (RDS)

3-tier Architecture Process Deployment

1. Starting with Route53 service, get a new domain and
2. Create a AMI for web tier and application tier.
3. Now create am SSL Certificate from AWS Certificate Manager to encrypt the traffic during transit.
4. Now create a VPC.
5. Create an Internet Gateway and attach it to VPC.
6. Start creating subnets for Load Balancer, Web Server in Web Tier.
7. Similarly create subnets for Load Balancer, App Server in Application tier.
8. To have redundant from zone failure, I have created two subnets in each availability zone for Load balancers and specific tier.
9. Create a subnets for Database as well.
10. Once all the subnets are created, create 2 route tables.
11. Create another subnet for Nat Gateway and associate it with Public Route table.
12. Create Nat Gateway to connect to internet from instances in private subnet.
13. Create Bastion Host in another VPC and create a VPC Peering connection between Bastion Host VPC and Production VPC.
14. Now create Security Groups to allow port 443 from Public.