* Can you provide the customer with a network diagram for the proposed AWS system that is in line with AWS best practices?

Same VPC as production VPC need to create for **dev, staging.**

Graphical user interface, application

Description automatically generated

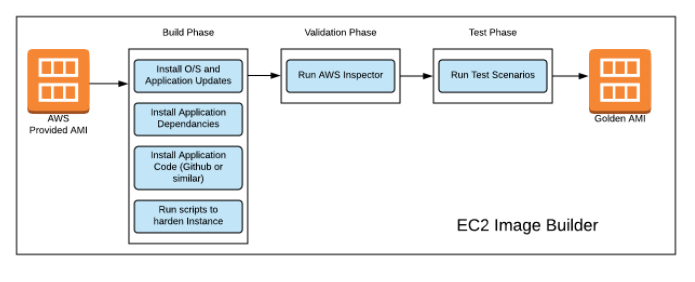
* Could you generate a template in either CloudFormation, Terraform, or similar IoC solution to create the proposed AWS infrastructure?

[terraform-provider-aws/examples at main · hashicorp/terraform-provider-aws · GitHub](https://github.com/hashicorp/terraform-provider-aws/tree/main/examples)

* **When the application is launched, a potentially large amount of traffic would be arriving in AWS. Could suggestions for improvements to the proposed solution be made to improve performance & resilience, while maintaining cost efficiency?**
  + **Use a Load Balancer – no matter how much the traffic increases, if you place your instances behind a Load Balancer it is always a great idea because the traffic is balanced across all the healthy instances.**
  + **Use an Auto Scaling Group – this that can scale up/down with as many instances as you want, this is a really powerful feature of AWS that one can use.**
  + **contact AWS to do an operation called “pre-warming”. What does that mean? This means that the AWS tech guys will configure the Load Balancer to have an appropriate level of capacity based on the expected traffic.**
* **The customer requires that AMI’s are tested in a staging account before they are shared with the production AWS account. How could this be done?**

**Ans:** This diagram borrowed from internet.

: [GitHub - masterwali/ec2-image-builder: A Terraform module that creates and manages an Amazon Machine Image (AMI) with EC2 Image Builder](https://github.com/masterwali/ec2-image-builder)



It is recommended

You can have your own pipeline Jenkins/concourse/ gitlabCI and Packer , ansible, chef tools integrate which will build your golden AMI in test Dev account and then can be shared to another account. You can force IAM police to use golden aim only.

* **The customer is looking to create a shared account that will service their dev, staging, and production accounts. The IP addresses in these regions overlap, so how could this be resolved?**

Ans:

* + **As this new migration from on premise to clouds I do not understand the reason behind ip overlap. While creating VPC you select different CIDR so there will be no overlap.**
  + **Network Address Translation (NAT) is a common technique to connect two networks with overlapping CIDRs.**
  + **The connectivity between these two networks need to be encrypted with IPSec.**
  + **The solution is to build a IPSEC tunnel between VPC-1 and VPC-2 and apply both source NAT (SNAT) and destination NAT (DNAT) n VPC-1 gateway**
  + **There are some tool available in AWS marketplace such as Aviatrix Systems.**
  + **As customer need 3 separate environment you can create 3 separate VPC for each environment . there is no need to create VOC peering in between VPS as these are separate environment and you can use CI/CD along with ICA so there is no ned of connectivity for data copy or code proportion. It can be manged through git and in work case you can have S3 to access the data across VPC.**
* **The customer has been using an Oracle database on-premises and are looking to save on licensing costs, could this be done?**

Ans: **This can be done with PostgreSQL**. **POstgreSQL Inherent Advantages Over Oracle Include**

* + **PostgreSQL is opensource and has no licensing cost, compared to per processor-based / user-based license for Oracle database.**
  + **No operational overhead of audits for licenses being used.**
  + **No expensive add-ons like Oracle as a wide variety of tools and extensions are available for PostgreSQL from community and third-parties.**

Tools :

* **AWS DMS** - [AWS Data Migration Service](https://docs.aws.amazon.com/dms/) (AWS DMS) helps you migrate your data to and from widely used commercial and open-source databases, including Oracle, MySQL, and PostgreSQL. The service supports homogeneous migrations such as Oracle to Oracle, and heterogeneous migrations such as Oracle to PostgreSQL or MySQL to Oracle.
* **AWS SCT -**[**AWS Schema Conversion Tool**](https://docs.aws.amazon.com/SchemaConversionTool/latest/userguide/CHAP_Welcome.html)**(AWS SCT) makes heterogeneous database migrations easy by automatically converting the source database schema and a majority of the custom code to a format compatible with the target database.**

Diagram

Description automatically generated

* **The web application writes a large amount of data to the database When the traffic increases, what changes or additions could be made so that application performance can be maintained or improved?**
  + **To support application performance whenever we can use ASG group for application server so the traffic can be managed by scaling number of instance and can be reduced number of instance whence load reduced, ilso it advisable to use network load balancer so the blue green deployment can be managed using route53 Weighted routing policy.**
  + **Many applications, including those built on modern**[**serverless architectures**](https://aws.amazon.com/serverless/)**, can have a large number of open connections to the database server, and may open and close database connections at a high rate, exhausting database memory and compute resources. Amazon RDS Proxy allows applications to pool and share connections established with the database, improving database efficiency and application scalability. With RDS Proxy, failover times for Aurora and RDS databases are reduced by up to 66% and database credentials, authentication, and access can be managed through integration with AWS Secrets Manager and AWS Identity and Access Management (IAM).**
  + **We can configure warm\_pool In ASG which Specifies the minimum number of instances to maintain in the warm pool. This helps you to ensure that there is always a certain number of warmed instances available to handle traffic spikes. Defaults to 0 if not specified.**