The Jenkins server will run 2 pipelines, one for infrastructure and the other for application.

Infrastructure: once the infrastructure pipeline is triggered, Jenkins builds the infrastructure and this is what it looks like; a custom vpc with 2 availability zones. 1 public and 1 private subnet in each availability zone. In the public subnets, we will have SonarQube server, and Nexus server, we will also have a bastion autoscaling group and a NAT gateway. In the private subnets, we will have 2 autoscaling group, one for stage and the other for production, the application requires a database and this is achieved using amazon RDS with multi-az enabled for high availability, also in the private subnet, we will also have the ansible which will be responsible for deploying the application on the docker host servers and managing dynamic inventory for deployment incase of scaling in and scaling out of the prod and stage autoscaling group. There will also be an S3 bucket which will house both the terraform statefile for Jenkins and vault server (which will be existing before the other infrastructure, i.e. not part of the pipeline) and the statefile for the other infrastructure within the pipeline. The vault will house the key management for the application servers which are docker-host as the application is containerized. The vault will also manage the authentication of the RDS database.

Application pipeline: Jenkins take the application code through the SonarQube and it is scanned for static code analysis using some quality gates that have been set by the quality and security team, once that is passed, it is taken through Maven( Jenkins plug in) for building, once built, the artifact is sent to Nexus, nexus will have 2 repository, maven and docker, the artefact is stored in the maven repository of nexus before Jenkins now builds a docker image with the same artefact, this artifact is then taken through Trivy for scanning, Trivy scans the base image and ensure it is free of known vulnerabilities. After the scan, Jenkins now stores the image in Nexus at the Docker repository. Jenkins then needs to connect to the bastion through SSM of the bastion and then create an SSH tunnel to Ansible since it is in the private subnet. Jenkins now triggers an Ansible playbook which ansible executes on the stage environment and production environment. The playbook instruction is for ansible to connect to nexus, fetch the docker image, deploy it on the docker host servers then create a container using the same image.