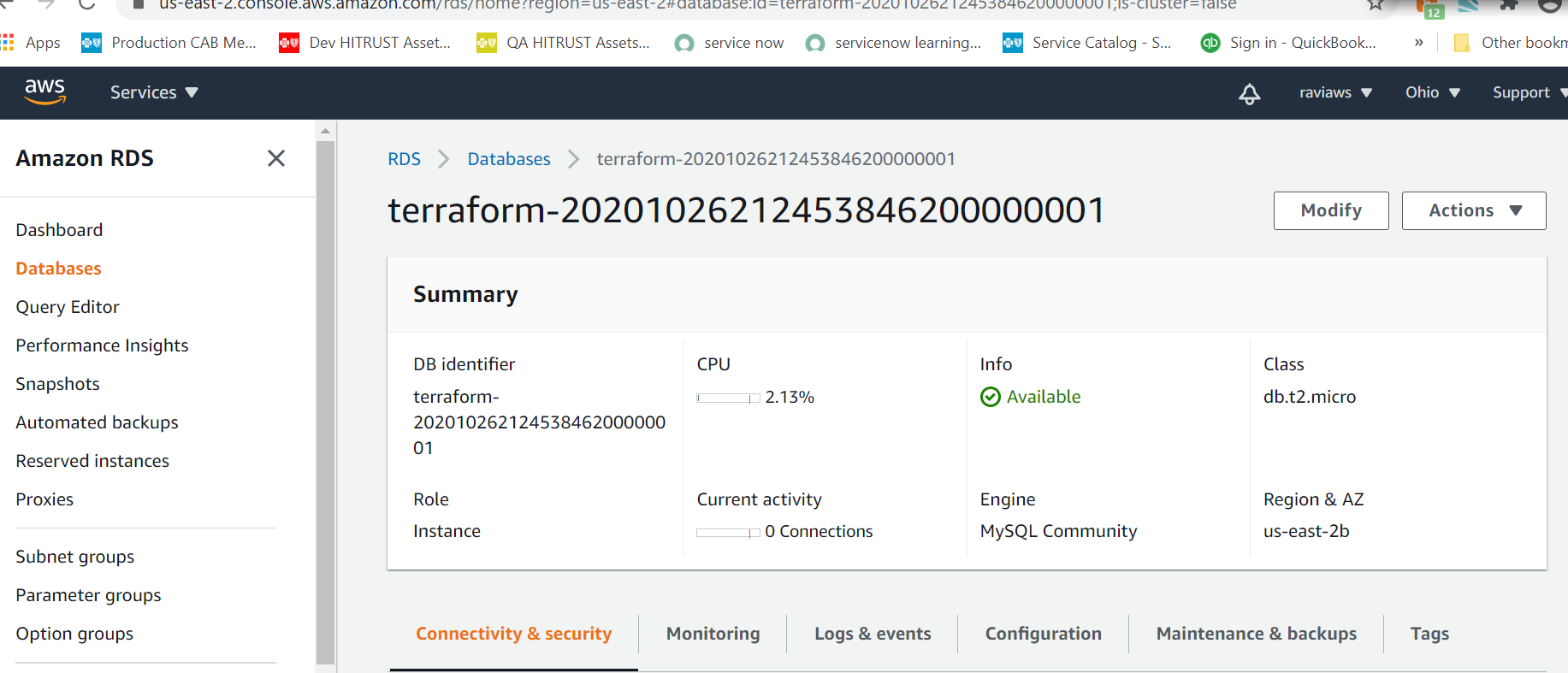
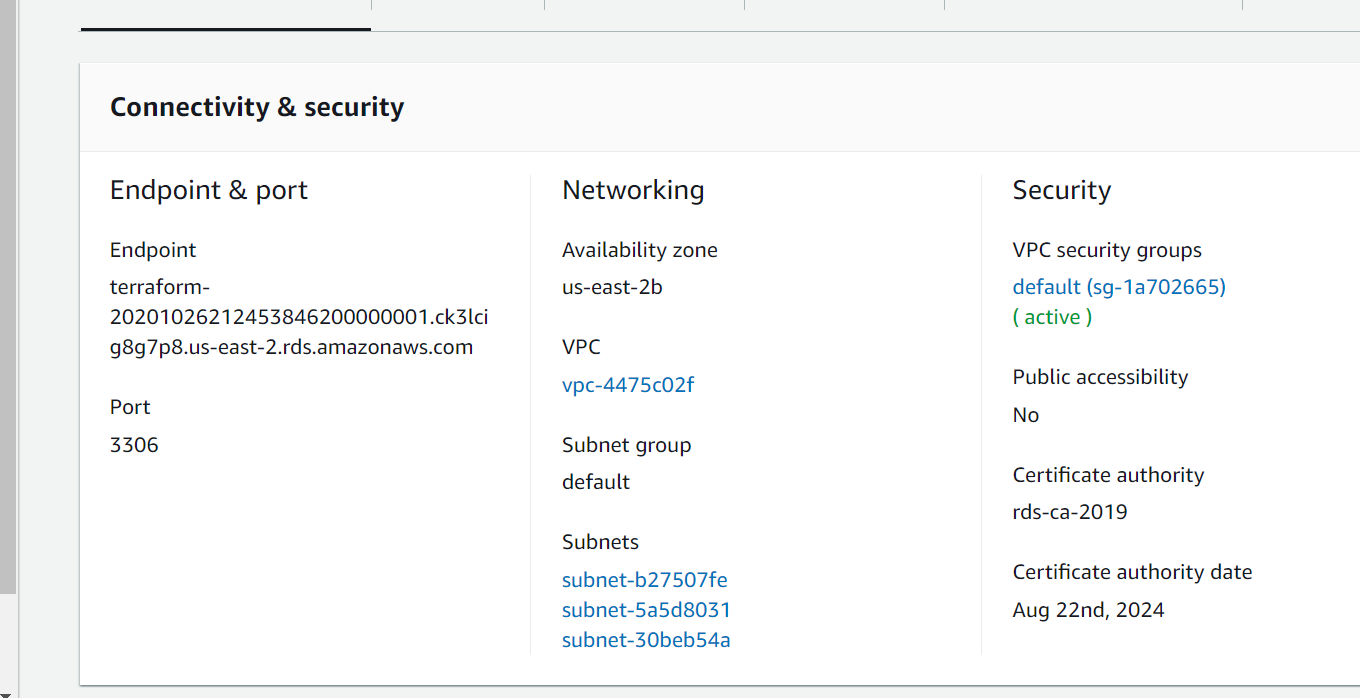
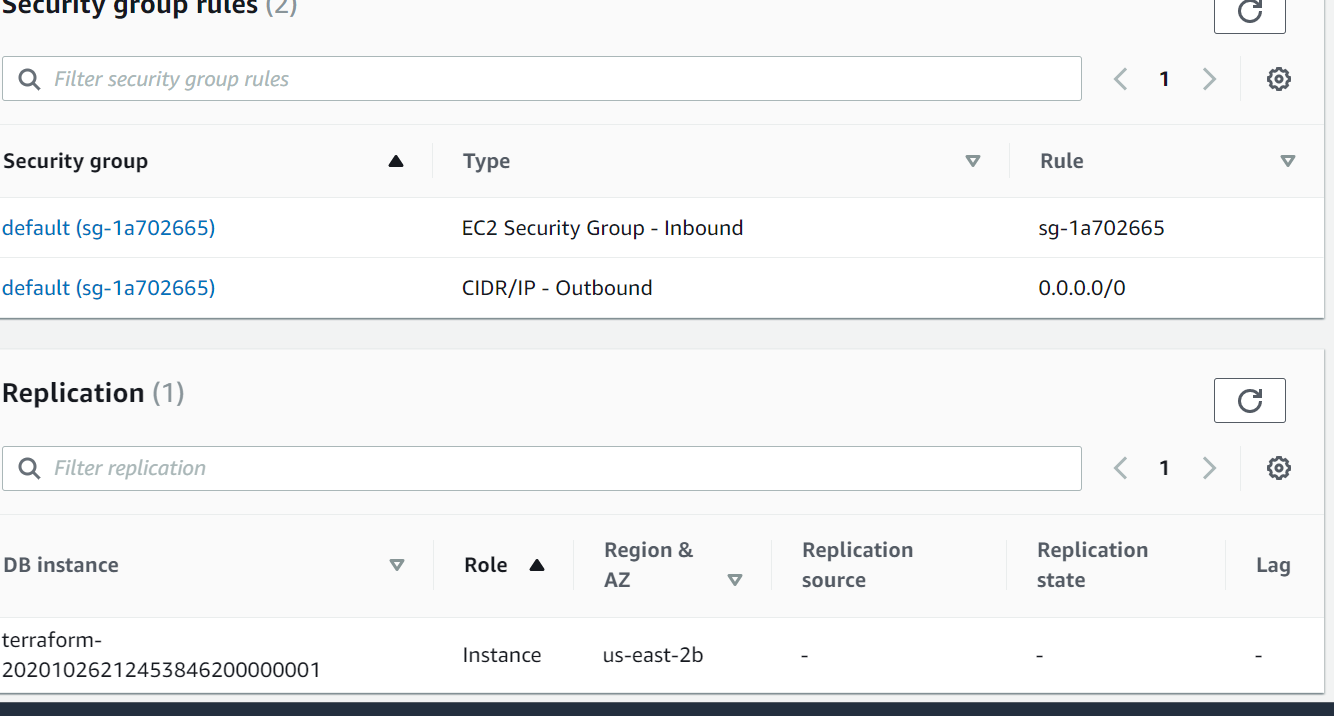
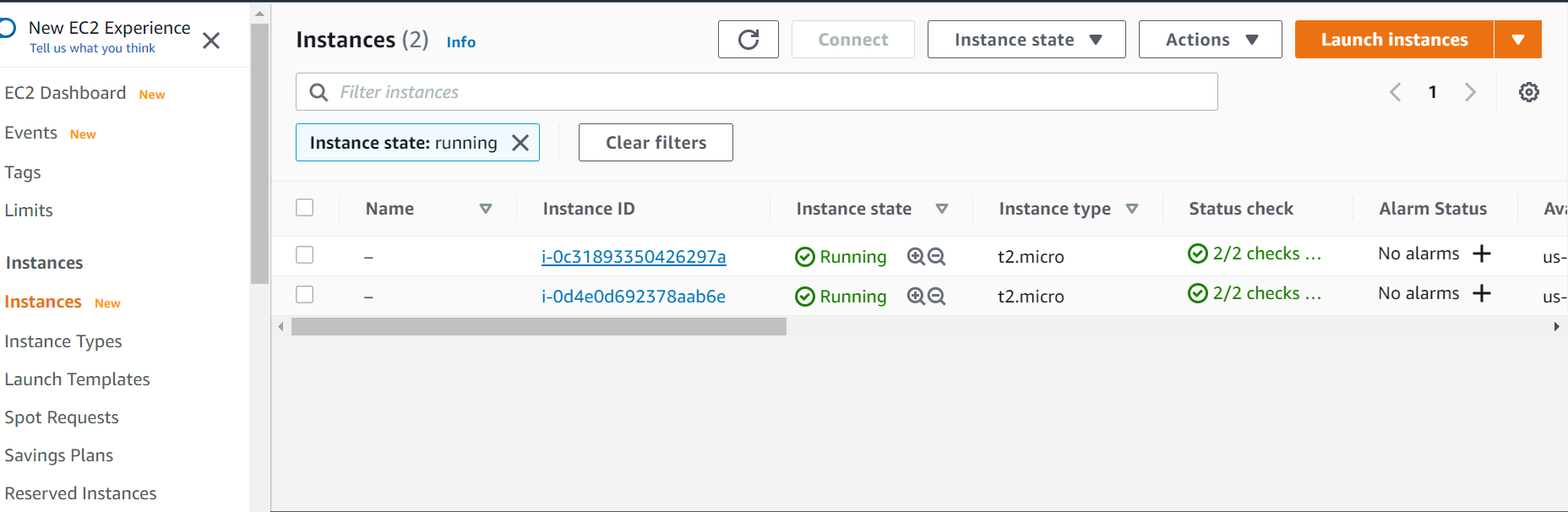
RDS info



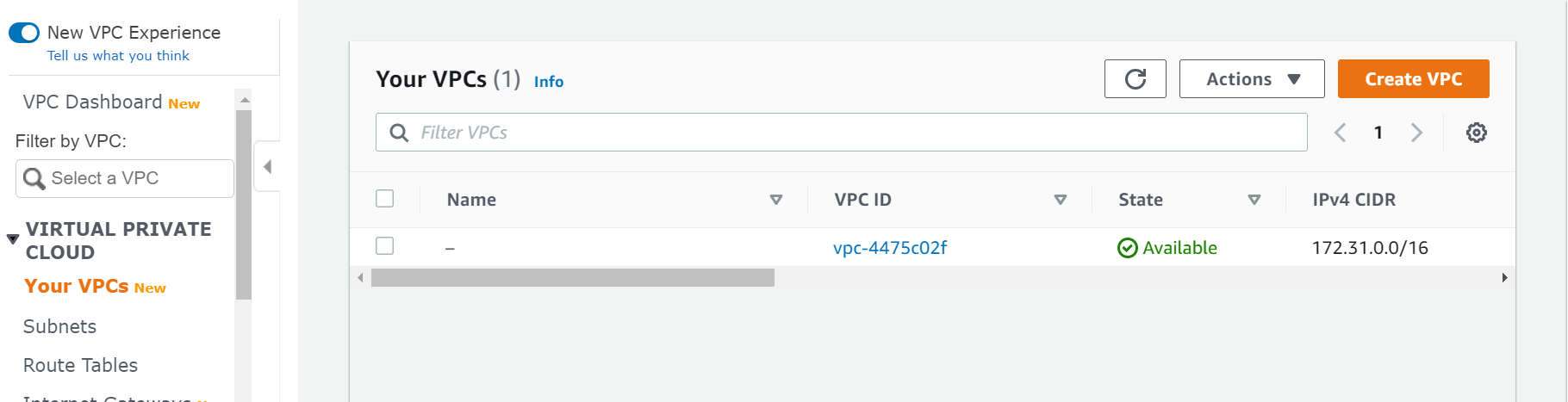




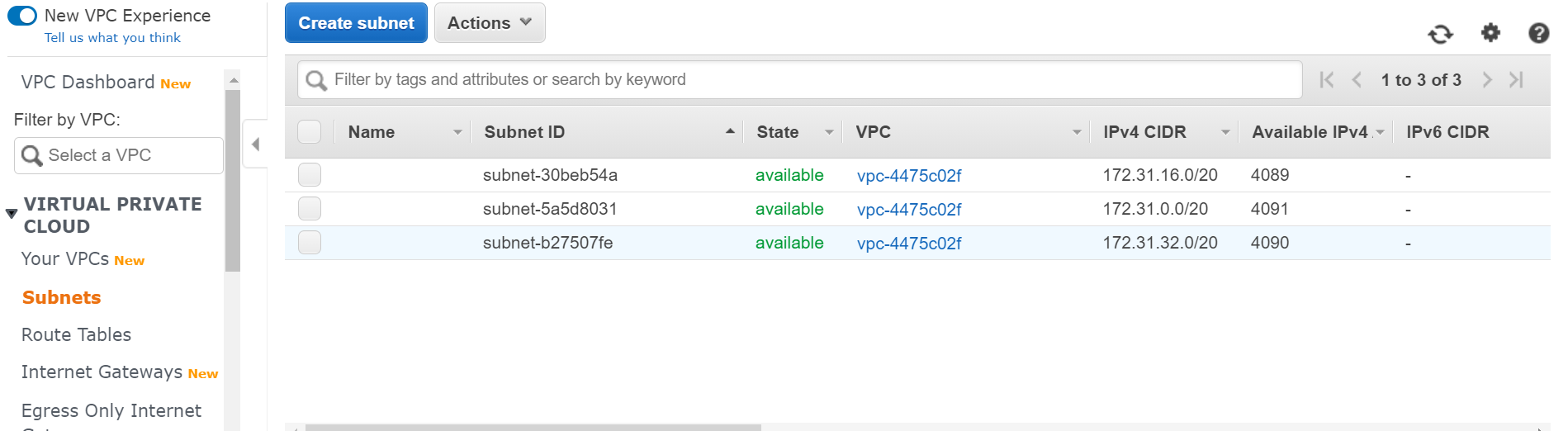
EC2 running Instance info.



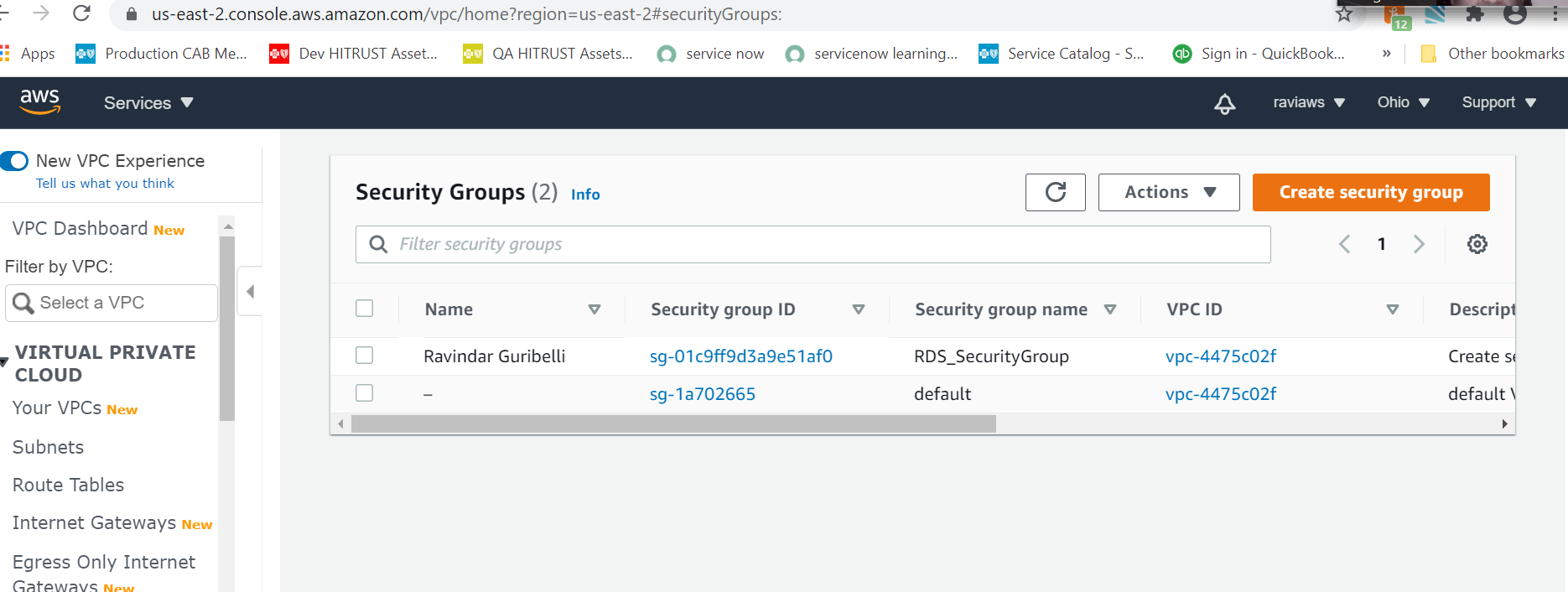
VPC info.



Subnet info.



Security groups for this instance.



Document any assumptions or additions you make in the README for the code repository. You may also want to consider and make some notes on:

* How would a future deployment know where to obtain the load balancer’s DNS name if it wanted to use this service?
* What aspects need to be considered to make the code work in a CD pipeline (how does it successfully and safely get into production)?

Each Classic Load Balancer receives a default Domain Name System (DNS) name. This DNS name includes the name of the AWS Region in which the load balancer is created. For example, if you create a load balancer named my-loadbalancer in the US West (Oregon) Region, your load balancer receives a DNS name such as my-loadbalancer-1234567890.us-west-2.elb.amazonaws.com. To access the website on your instances, you paste this DNS name into the address field of a web browser. However, this DNS name is not easy for your customers to remember and use.

If you'd prefer to use a friendly DNS name for your load balancer, such as www.example.com, instead of the default DNS name, you can create a custom domain name and associate it with the DNS name for your load balancer. When a client makes a request using this custom domain name, the DNS server resolves it to the DNS name for your load balancer.

It's easier to move your project to a different CI server, since most of the CI configuration does not need to be re-created on the new server since it is in the code repo. With a web UI, you end up spending a lot of time clicking around trying to make the new server's job configuration look the same as the old server's, copying-and-pasting various bits of text, and it's easy to miss something.