To the DevOps Team…

Many thanks for sending on this Tech test. I did not get to spend as much time on it as I would like, so please accept apologies in advance, for not completing all requirements.   
I really enjoyed delving into it, and hopefully what I have delivered is acceptable ☺

For your consideration:

* I previously informed Gary (Fox) that I had not yet worked with Kubernetes or EKS.   
  However, I researched into how k8s files & EKS clusters are built and configured to complete this assignment, though what is delivered are the files at their most basic.  
  **Note:** I did not attempt helm charts.
* I was keen to validate the Jenkins pipeline syntax, but a majority of my time was spent setting up my local environment and a Jenkins server & agent, as my home laptop was lacking. I have documented my steps in this, for your interest, in .\script\LocalSetup.txt.
* As I like to validate each component, I also tested the NodeJs app in isolation at various stages, before reaching a Jenkins build.
* Much time was lost resolving yamllint issues running on the Jenkins container.
* Finally, I ran into connectivity glitches between Jenkins and EKS cluster, and was unable to push the deployment, though I have still included the syntax in the final Jenkinsfile. Despite best efforts I suspect its still down to IAM permissions, but ran out of time to resolve.  
  **Note:** I did not complete the Unit tests.
* I added extra efforts into creating many of the AWS objects through Terraform, as seen in .\terraform – VPC, IAM, ECR, EKS.
* All references to AWS is my personal account, will need to be changed for any testing.

What I delivered:

* Git repo available at <https://github.com/bronandrews/NodeJs-Tech-Test> with desired folder structure, where applicable.
* Full AWS stack written in Terraform, with VPC networking, IAM users & permissions, ECR repo and EKS cluster.
* Jenkins pipeline that validates k8s yaml, builds a NodeJs app, pushes into an ECR repo and deploys k8s files to an EKS cluster.

Also:

* A Dockerfile that runs a Jenkins server & agent in local docker, with all plugins and packages required by the pipeline.

Many thanks again, this was an enjoyable task!  
  
Regards,

Bronwyn Andrews.