

Continuous integration



GitHub

- Docker File: container environment for API
- Docker File: container environment for SQL Database
- YAML File: SQL Database Docker container
- YAML File: API Docker container
- YAML File: CloudFormation template
- YAML File: ECS Task Definitions
- Xero-Refactored-App src code
- Database schema + environment scripts

Optionally

- Jenkins setup + plugins script
- Jenkins pipeline setup script



Branches

Merge



Master

Push code

Development team



- Compile src code
- Pull Docker file from Github
- Pull YAML files from Github
- Run YAML files
- Build Docker images
- Run Docker images
- Run Unit Tests
- Run Integration Tests
- Tear down test environment
- Build artifact

Feedback

Run Tests



Unit tests



Integration tests



Continuous deployment



Jenkins

- Push Docker images to AWS ECR
- Run CloudFormation template
- Run ECS Task Definition deployments
- Update Load Balancer rules

Deploy updated definitions

Push updated images

Run Template

aws AWS Cloud



CloudFormation



CloudWatch Alerts



Local Security Notifications



Notification



VPC



Elastic Container Registry

Elastic Container Service Cluster



Task 1

Definition: API

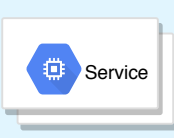


Task 2

Definition: SQL Database

Private Subnet

BLUE



Service



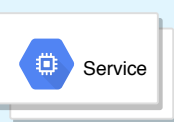
Nano



EC2 Instance

API App
API Endpoints

GREEN



Service



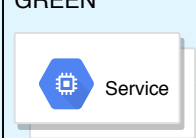
Nano



EC2 Instance

API App
API Endpoints

GREEN



Service



Small



EC2 Instance

Database
Microsoft SQL Database

BLUE



Service



Small



EC2 Instance

Database
Microsoft SQL Database

Listener 80

Listener 8080

Listener 8080

Listener 8080

Listener 80



Application Load Balancer

Blue-Green swap

Considerations:

- This tool chain can be simplified by using other tools that have a complete CI/CD + git version control setup, eg. GitLab. For the purpose of this demonstration focus will be around using GitHub and Jenkins.
- Using Blue-Green deployment model for releasing changes to both instances on ECS. This will cater for schema changes in the database or breaking changes in the app.
- Main Jenkins plugins used in this design: Blue Ocean, GitHub, GitHub Integration, Docker, Amazon EC2 Container Service, Pipeline: AWS Steps
- Cost estimation:

<div>Amazon EC2<div>Region: Asia Pacific (Sydney)</div><div><div>Edit</div><div>Action ▼</div></div></div> <div><div>Quick estimate</div><div>Operating system (Windows Server), Quantity (2), Storage for each EC2 instance (General Purpose SSD (gp2)), Storage amount (1 GB), Instance type (t2.nano)</div><div>Monthly:10.75 USD</div></div>		
<div>Amazon EC2<div>Region: Asia Pacific (Sydney)</div><div><div>Edit</div><div>Action ▼</div></div><div><div>Quick estimate</div><div>Operating system (Windows Server), Quantity (2), Storage for each EC2 instance (General Purpose SSD (gp2)), Storage amount (30 GB), Instance type (t3.small)</div><div>Monthly:60.20 USD</div></div></div>		
First 12 months total	Total upfront	Total monthly
851.40 USD	0.00 USD	70.95 USD

Optionally

<div>Services (4)</div> <div>Amazon CloudWatch<div>Region: Asia Pacific (Sydney)</div><div><div>Edit</div><div>Action ▼</div></div></div> <div><div>Monthly:1.71 USD</div></div>		
<div>Elastic Load Balancing<div>Region: Asia Pacific (Sydney)</div><div><div>Edit</div><div>Action ▼</div></div><div>Application Load Balancer</div><div>Number of Application Load Balancers (1)</div><div>Monthly:2,938.40 USD</div></div>		
First 12 months total	Total upfront	Total monthly
36,132.66 USD	0.00 USD	3,011.06 USD