

Project Proposal

What we have:

- A Github Repository that stores Python server source code that runs using Uvicorn

Objective:

- Deploy using Github Action with hooks and trigger.
- The Python API server on AWS need to have high availability and monitoring options.
- Ensure security and permission compliance.

Note:

As we only need to deploy a Python API server, I suggest we use ECS with Fargate that host containerization application rather than a whole EC2. This helps reduce cost, also have built-in auto scaling ability and reduce complexity in deploying and maintenance. The diagram is at the end of the proposal

Proposal with step by step:

- Setup trigger for Github Actions along with hooks.
- Use Github Actions to build an image from Python source code by writing a github workflow file.
- Deploy the image to ECR on trigger.
- Setup Amazon RDS with Multi-AZ option on for high availability.
- Setup ECS with auto scaling configuration and tasks including environment variable that will be used inside the image with the image path pointing to ECR.
- Setup the same VPC for both services with suitable IAM role on ECS for connection to Amazon RDS.
- If Fargate is still running while new image is uploaded, it will continue running (can use --force-new-update in ECS).
- Use ALB along with API gateway and Route 53 for connection. ALB is for high availability, API gateway for connection, Route 53 for domain settings
- For monitoring and logging, turn on logging options for ALB for network debug, configure ECS containers to download Cloudwatch agent on startup, use Cloudwatch Events that trigger on specific errors from container with destination as an SNS service that will send an email to a list of subscribers.
- For Permission and Security, depending on organization structure, can use services ranging from AWS Service Catalog to AWS Organizations, use AWS Config for permission compliance

Diagram

