Provisioning Dev Container on AWS Cloud

Prerequisites

1. AWS Account with an IAM User with administrative permissions.
2. Docker Desktop installed.
3. Knowledge of launching EC2 instances.

Steps

EC2 Instance

1. Log in to your AWS Account and open EC2 Service.
2. Launch the EC2 instance with the following attributes:
    - Use Linux AMI
    - Attach Key pair(login)
    - Attach security group with allowing SSH traffic rule
    - User Data script,

```
#!bin/sh
sudo yum update
sudo yum install -y docker
sudo usermod -a -G docker ec2-user
newgrp docker
sudo systemctl start docker.service

sudo yum update
sudo curl -L
https://github.com/docker/compose/releases/latest/download/docker-
compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

This script installs Docker & Docker Compose on the instance required to run Dev Container.

Local Docker Setup

1. Start the Docker Desktop.

2. Open a Powershell window.

3. Run the following commands to create a docker context for the respective EC2 instance:

    - `docker context create "context-name" --docker host=ssh://ec2-user@"instance-ip"`

4. Next, we will switch from the default context (docker locally) to the above-created context by running:
    - `docker context use "context-name"`

    Substitute the *context-name* & *instance-ip* with the custom context name and the above provisioned EC2 instance public ip respectively.
    This will let us run the Dev Container onto the EC2 instance.

5. Then in the same terminal window, check whether the Docker CLI can communicate with docker on the EC2 instance by running:
    - `docker run --rm hello-world`.

> If you cannot run the container, check out Visual Studio Code's SSH Tunneling for Docker guide to help you troubleshoot connection issues.

5. If you can successfully run the above command, you can start running dev containers on the remote machine by referring to the following document.
    - Dev Container Setup
