To manage RabbitMQ users and set up policies in your Dockerized RabbitMQ environment, follow these detailed steps:

### Step 1: Setting Up RabbitMQ Environment

Assuming you already have a RabbitMQ cluster running with Docker Compose (one master and two slave nodes), make sure your environment is up and running:

```bash

docker-compose up -d

```

### Step 2: Access RabbitMQ Management Plugin

Ensure the RabbitMQ Management Plugin is enabled on all nodes. You can access it via the browser:

```plaintext

http://localhost:15672

```

### Step 3: Adding a User

1. \*\*Login to RabbitMQ Management UI\*\*:

- Go to `http://localhost:15672`.

- Log in with default credentials: `guest/guest`.

2. \*\*Create a New User\*\*:

- Navigate to the \*\*"Admin"\*\* tab.

- In the "Add a user" section, provide the following:

- \*\*Username\*\*: The desired username.

- \*\*Password\*\*: The desired password.

- \*\*Tags\*\*: User tags (e.g., `administrator`, `management`, etc.).

- Click \*\*"Add user"\*\*.

3. \*\*Set User Permissions\*\*:

- After creating the user, you need to set permissions.

- Click on the newly created user from the list.

- In the \*\*Permissions\*\* section, set the following:

- \*\*Configure\*\*: Regex for configuring resources (e.g., `.\*` for all).

- \*\*Write\*\*: Regex for resources the user can write to.

- \*\*Read\*\*: Regex for resources the user can read.

- Click \*\*"Set permissions"\*\*.

### Step 4: Creating and Applying Policies

Policies in RabbitMQ help manage features like message TTL, mirrored queues, etc. Here’s how to create and apply a policy:

1. \*\*Navigate to Policies\*\*:

- In the RabbitMQ Management UI, go to the \*\*"Admin"\*\* tab.

- Click on \*\*"Policies"\*\*.

2. \*\*Add a New Policy\*\*:

- Click on \*\*"Add / update a policy"\*\*.

- Provide the following details:

- \*\*Name\*\*: Give a name to your policy (e.g., `mirror-policy`).

- \*\*Pattern\*\*: Regex pattern that matches queue names (e.g., `^flight.\*` to match queues starting with `flight`).

- \*\*Apply to\*\*: Choose `Exchanges` or `Queues`.

- \*\*Definition\*\*: Provide the policy details, for example:

```json

{"ha-mode":"exactly","ha-params":2,"ha-sync-mode":"automatic"}

```

- \*\*Priority\*\*: Set the priority for the policy.

3. \*\*Apply the Policy\*\*:

- Click \*\*"Add policy"\*\*.

- The policy will be applied to the queues/exchanges that match the pattern.

### Step 5: Command-Line Management (Optional)

If you prefer managing users and policies via command-line, use the following commands within your Docker container:

1. \*\*Exec into the RabbitMQ container\*\*:

```bash

docker exec -it <rabbitmq-container-name> /bin/bash

```

2. \*\*Add a user\*\*:

```bash

rabbitmqctl add\_user myuser mypassword

rabbitmqctl set\_user\_tags myuser administrator

```

3. \*\*Set user permissions\*\*:

```bash

rabbitmqctl set\_permissions -p / myuser ".\*" ".\*" ".\*"

```

4. \*\*Add a policy\*\*:

```bash

rabbitmqctl set\_policy -p / mirror-policy "^flight.\*" '{"ha-mode":"exactly","ha-params":2,"ha-sync-mode":"automatic"}' --priority 1

```

### Step 6: Verify Configuration

- \*\*Check User Permissions\*\*:

- Ensure that the user has the appropriate access by testing access to queues or exchanges.

- \*\*Verify Policies\*\*:

- Ensure that the policies are applied correctly by inspecting the relevant queues or exchanges.

By following these steps, you can effectively manage RabbitMQ users and policies in your Dockerized RabbitMQ environment.