ElastiCache Stresser – Phase 3: Automated Alarm Validation

This phase introduces automation for validating AWS CloudWatch alarm behavior after stress tests. It ensures that alarms are triggered and recovered correctly based on performance thresholds defined in Phase 2.

1. Goal

Automate CloudWatch alarm state checks to verify that alarms behave as expected during and after CacheStressTester runs. Optionally integrate with Slack, Datadog, or EventBridge for notifications and reporting.

2. Scope

• Detect CloudWatch alarm transitions (OK → ALARM → OK). • Log state updates for evidence. • Optionally notify external channels (Slack, Datadog, etc.).

3. Components

1. **CacheStressTester** – continues generating Redis load. 2. **CloudWatch Checker** – new module/script using AWS SDK or CLI to list alarm states. 3. **Validator** – correlates alarm state timestamps with CacheStressTester execution logs.

4. Implementation Example (C# AWS SDK)

Example for fetching active alarms programmatically: ``` var client = new AmazonCloudWatchClient(); var response = await client.DescribeAlarmsAsync(new DescribeAlarmsRequest { StateValue = StateValue.ALARM }); foreach (var alarm in response.MetricAlarms) { Console.WriteLine($"{alarm.AlarmName} - {alarm.StateValue} at {alarm.StateUpdatedTimestamp}"); } ```

AWS CLI Alternative:

``` aws cloudwatch describe-alarms --state-value ALARM ``` Use CacheStressTester logs to align alarm state changes with load test intervals.

5. Automation Scenarios

Scenario	Expected Result
High load (≥ 500 threads)	HighCPU-RedisCluster enters ALARM
Moderate load (≤ 300 threads)	Alarm remains in OK
Cooldown phase	Alarm returns to OK within 5 min

6. Optional Extensions
```

• Export validation results to CSV/JSON for QA documentation. • Send summary messages to a Slack webhook. • Use Datadog API to confirm mirrored alarms. • Configure an AWS EventBridge rule to trigger Lambda when alarm state = ALARM.

7. Next Deliverable

Develop a lightweight CLI tool or PowerShell script (`AlarmValidator`) that: • Accepts alarm names, tags, and time window as input. • Retrieves CloudWatch alarm states and evaluates pass/fail results. • Outputs structured JSON or console logs for automated test reporting.

References

• AWS SDK for .NET: https://docs.aws.amazon.com/sdk-for-net/ • CloudWatch DescribeAlarms API: https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/API_DescribeAlarms.html • AWS CLI Reference: https://docs.aws.amazon.com/cli/latest/reference/cloudwatch/describe-alarms.html • EventBridge Triggers: https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-create-rule.html