Monitoring & Alerting Checklist

Introduction

Effective monitoring and alerting are critical for maintaining the reliability, availability, and security of your cloud infrastructure. This checklist outlines core practices for setting up observability using AWS-native tools, open-source platforms like Prometheus and Grafana, and integrations with Slack or Opsgenie for incident response.

1. Define What to Monitor

Identify components critical to business operations:

- EC2 instances (CPU, memory, disk usage, network)
- RDS/Aurora databases (connections, IOPS, storage, replication lag)
- EKS clusters (pod health, node utilisation, kubelet status)
- Load balancers (latency, HTTP error rates, request count)
- Application-level metrics (latency, error rates, queue length)

2. Establish Baselines and Thresholds

Use historical performance data to set normal operating ranges.

Tips:

- Use AWS CloudWatch metrics history
- Configure Prometheus to track trends
- Establish warning and critical thresholds

3. Create Detailed Dashboards

Use Grafana or CloudWatch Dashboards:

- Environment-specific dashboards
- Group by service or resource type
- Colour-coded indicators
- Display current state and trends

4. Set Up Actionable Alerts

Alerts must be specific, timely, and actionable.

Monitoring & Alerting Checklist

Use CloudWatch Alarms or Prometheus Alertmanager. Integrate with Slack, Teams, or Opsgenie.

Avoid alert fatigue.

Example Terraform:

```
resource "aws_cloudwatch_metric_alarm" "high_cpu" {
  alarm_name = "HighCPUUtilisation"
  comparison_operator = "GreaterThanThreshold"
  evaluation_periods = 2
  metric_name = "CPUUtilization"
  namespace = "AWS/EC2"
  period = 120
  statistic = "Average"
  threshold = 80
  alarm_description = "Alarm when CPU exceeds 80%"
  dimensions = {
    InstanceId = aws_instance.my_instance.id
  }
}
```

5. Centralise Logs

Collect and store logs centrally:

- CloudWatch Logs
- Fluent Bit with Amazon OpenSearch
- Loki + Grafana for EKS
- S3 for retention

Set log retention policies to manage cost.

6. Monitor DNS, SSL and External Endpoints

- Use Route 53 Health Checks or UptimeRobot
- Monitor SSL expiry with Certbot scripts or Lambda
- Alert on response time thresholds

Monitoring & Alerting Checklist

7. Audit IAM and Security Events

Security monitoring:

- Monitor IAM changes with CloudTrail
- Track failed login attempts
- Watch GuardDuty alerts
- Audit root account usage

8. Review and Iterate

Monitoring is not set-and-forget:

- Remove noisy alerts
- Update thresholds
- Conduct post-mortems
- Improve alert logic

Conclusion

Monitoring and alerting improves awareness, incident response, and system resilience. Combine AWS-native tools, third-party platforms, and automation to keep systems secure and healthy.

Visit https://www.adesoji.dev for Terraform examples and templates.