**AWS EC2 Monitoring Runbook**

**Table of Contents**

1. **Introduction**
   * 1.1 [Purpose of the Document](#11-purpose-of-the-document)
   * 1.2 [Scope](#12-scope)
2. **Monitoring Overview**
   * 2.1 [Importance of Monitoring EC2 Instances](#21-importance-of-monitoring-ec2-instanc)
   * 2.2 [Establishing Performance Baselines](#22-establishing-performance-baselines)
   * 2.3 [Monitoring Tools and Services](#23-monitoring-tools-and-services)
3. **Metrics to Monitor**
   * 3.1 [CPU Metrics](#31-cpu-metrics)
   * 3.2 [Network Metrics](#32-network-metrics)
   * 3.3 [Disk Metrics](#33-disk-metrics)
   * 3.4 [Status Checks](#34-status-checks)
   * 3.5 [EBS Metrics](#35-ebs-metrics)
   * 3.6 [Memory Metrics (Requires CloudWatch Agent)](#36-memory-metrics-requires-cloudwatch-a)
4. **Monitoring Guidelines**
   * 4.1 [CPU Utilization Monitoring](#41-cpu-utilization-monitoring)
   * 4.2 [Network Monitoring](#42-network-monitoring)
   * 4.3 [Disk Monitoring](#43-disk-monitoring)
   * 4.4 [Status Checks Monitoring](#44-status-checks-monitoring)
   * 4.5 [EBS Metrics Monitoring](#45-ebs-metrics-monitoring)
   * 4.6 [Memory Monitoring](#46-memory-monitoring)
5. **Commands for Monitoring on Amazon Linux, RHEL 9, and Ubuntu 24**
   * 5.1 [CPU Utilization Commands](#51-cpu-utilization-commands)
   * 5.2 [Memory Usage Commands](#52-memory-usage-commands)
   * 5.3 [Disk Usage Commands](#53-disk-usage-commands)
   * 5.4 [Disk I/O Commands](#54-disk-io-commands)
   * 5.5 [Network Usage Commands](#55-network-usage-commands)
   * 5.6 [System Logs Commands](#56-system-logs-commands)
   * 5.7 [CloudWatch Agent Status Commands](#57-cloudwatch-agent-status-commands)
6. **Troubleshooting Common Issues**
   * 6.1 [High CPU Utilization](#61-high-cpu-utilization)
   * 6.2 [High Memory Usage](#62-high-memory-usage)
   * 6.3 [Disk I/O Performance Issues](#63-disk-io-performance-issues)
   * 6.4 [Network Issues](#64-network-issues)
   * 6.5 [Status Check Failures](#65-status-check-failures)
   * 6.6 [CloudWatch Agent Issues](#66-cloudwatch-agent-issues)
7. **Best Practices**
   * 7.1 [Setting Realistic Thresholds](#71-setting-realistic-thresholds)
   * 7.2 [Reducing False Positives](#72-reducing-false-positives)
   * 7.3 [Regularly Reviewing Metrics and Alarms](#73-regularly-reviewing-metrics-and-alar)
   * 7.4 [Automating Responses](#74-automating-responses)
8. **References**

1. Introduction

1.1 Purpose of the Document

This runbook provides a comprehensive guide for monitoring Amazon EC2 instances running on Amazon Linux, Red Hat Enterprise Linux 9 (RHEL 9), and Ubuntu 24. It outlines key metrics to monitor, commands to check system health, specific monitoring guidelines for each metric, and troubleshooting steps for common issues.

1.2 Scope

The document focuses on EC2 instances and covers the following:

* Key metrics available via Amazon CloudWatch
* Commands for monitoring on Amazon Linux, RHEL 9, and Ubuntu 24
* Troubleshooting steps for common issues affecting EC2 instances
* Best practices for effective monitoring

| **Metric** | **Threshold** | **Priority** | **Source** |
| --- | --- | --- | --- |
| CPU Utilization | >80% for 5 minutes | High | CloudWatch |
| Memory Usage | >85% for 5 minutes | High | CloudWatch Agent |
| Disk Usage | >85% | Critical | CloudWatch Agent |
| Network I/O | Anomaly detection | Medium | CloudWatch |
| Status Checks | Any failure | Critical | CloudWatch |

2. Monitoring Overview

2.1 Importance of Monitoring EC2 Instances

Effective monitoring ensures that applications run smoothly, performance bottlenecks are identified early, and resources are utilized efficiently. It enables proactive detection and resolution of issues before they impact end-users.

OS-Specific Monitoring

Amazon Linux

System Commands

# Resource Monitoring

top -b -n 1 # CPU and memory snapshot

vmstat 1 5 # Virtual memory statistics

free -h # Memory usage

df -h # Disk usage

iostat -xz 1 5 # Disk I/O statistics

netstat -tulpn # Network connections

# Log Locations

/var/log/messages # System logs

/var/log/amazon/ssm/ # SSM agent logs

/var/log/cloud-init.log # Instance initialization

# Performance Analysis

sar -u 1 5 # CPU utilization

sar -r 1 5 # Memory utilization

sar -n DEV 1 5 # Network statistics

**2.2 RHEL 9**

**System Commands**

# System Status

systemctl status # Service status

journalctl -xe # System logs

dnf check-update # Available updates

# Resource Monitoring

top -b -n 1

free -m

df -hT # Include filesystem type

pidstat 1 5 # Per-process statistics

# Security

firewall-cmd --list-all # Firewall rules

semanage port -l # SELinux port mappings

**2.3 Ubuntu 24**

**System Commands**

# System Status

systemctl status

journalctl -f # Follow system logs

apt list --upgradable # Available updates

# Resource Monitoring

htop # Interactive process viewer

iotop # I/O monitoring

nmon # Performance monitoring

lsof -i # Network connections

**3. Troubleshooting Procedures**

**3.1 High CPU Usage**

1. Identify top processes:

ps aux --sort=-%cpu | head -10

top -b -n 1 | head -20

1. Check system load:

uptime

sar -u 1 10

1. Resolution Steps:

* Analyze application logs
* Check for resource-intensive cron jobs
* Consider vertical scaling
* Review application optimization

**3.2 Memory Issues**

1. Check memory status:

free -h

vmstat 1 5

cat /proc/meminfo

1. Analyze swap usage:

swapon --show

cat /proc/swaps

1. Resolution Steps:

* Clear page cache if needed:

sync; echo 1 > /proc/sys/vm/drop\_caches

* Adjust application memory limits
* Check for memory leaks
* Consider instance resize

**3.3 Disk Space Issues**

1. Identify large files:

du -sh /\* | sort -hr

find / -type f -size +100M -exec ls -lh {} \;

1. Check inode usage:

df -i

find / -xdev -printf '%h\n' | sort | uniq -c | sort -k 1 -n

1. Resolution Steps:

* Remove old logs
* Clean package cache:

# Amazon Linux/RHEL

sudo yum clean all

# Ubuntu

sudo apt-get clean

* Extend EBS volume
* Archive old data

**3.4 Network Issues**

1. Check connectivity:

ping -c 4 8.8.8.8

traceroute google.com

curl -v telnet://destination:port

1. Network statistics:

netstat -s

ss -neita

iptables -L -v -n

1. Resolution Steps:

* Verify security groups
* Check route tables
* Validate DNS resolution
* Review network ACLs

**4. Preventive Maintenance**

**4.1 Daily Tasks**

1. Monitor logs:

sudo tail -f /var/log/messages # Amazon Linux/RHEL

sudo tail -f /var/log/syslog # Ubuntu

1. Check disk space:

df -h

ncdu / # If installed

1. Verify services:

systemctl list-units --state=failed

**4.2 Weekly Tasks**

1. System updates:

# Amazon Linux/RHEL

sudo yum update -y

# Ubuntu

sudo apt update && sudo apt upgrade -y

1. Security checks:

sudo aureport -au # Auth attempts

sudo fail2ban-client status # If fail2ban installed

**2.2 Establishing Performance Baselines**

Before setting up monitoring and alarms, it's crucial to establish performance baselines for your EC2 instances. Baselines represent normal operating conditions and help in defining realistic thresholds for alerts.

**2.3 Monitoring Tools and Services**

* **Amazon CloudWatch**: Provides monitoring and observability of AWS resources and applications.
* **CloudWatch Agent**: Installed on instances to collect additional system-level metrics.

# Amazon Linux/RHEL

sudo yum install -y amazon-cloudwatch-agent

# Ubuntu

sudo apt-get update

sudo apt-get install -y amazon-cloudwatch-agent

# Configure and start agent

sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard

sudo systemctl start amazon-cloudwatch-agent

* **AWS Observability Best Practices**: Recommendations for setting up effective monitoring.

**3. Metrics to Monitor**

**3.1 CPU Metrics**

**CPUUtilization**

* **Description**: Measures the percentage of allocated EC2 compute units currently in use on the instance.
* **Importance**: High CPU utilization may indicate that an instance is overburdened and could lead to performance issues.

**3.2 Network Metrics**

**NetworkIn and NetworkOut**

* **NetworkIn**: The number of bytes received on all network interfaces by the instance.
* **NetworkOut**: The number of bytes sent out on all network interfaces by the instance.
* **Importance**: Monitoring these metrics helps detect network bottlenecks, unusual traffic patterns, or potential attacks.

**3.3 Disk Metrics**

**DiskReadOps and DiskWriteOps**

* **DiskReadOps**: Count of completed read operations from all instance store volumes available to the instance.
* **DiskWriteOps**: Count of completed write operations to all instance store volumes available to the instance.
* **Importance**: High values may indicate disk I/O bottlenecks affecting application performance.

**DiskReadBytes and DiskWriteBytes**

* **DiskReadBytes**: Bytes read from all instance store volumes.
* **DiskWriteBytes**: Bytes written to all instance store volumes.
* **Importance**: Helps in understanding data throughput and detecting abnormal disk usage patterns.

**3.4 Status Checks**

**StatusCheckFailed**

* **Description**: Reports whether the instance has passed both the instance status check and the system status check (0: passed, 1: failed).
* **Importance**: Identifies if the instance or the underlying system has issues affecting performance or availability.

**StatusCheckFailed\_Instance**

* **Description**: Reports whether the instance has passed the instance status check.
* **Importance**: Detects issues like exhausted memory, corrupted file system, or incorrect network configurations.

**StatusCheckFailed\_System**

* **Description**: Reports whether the instance has passed the system status check.
* **Importance**: Indicates problems with AWS infrastructure that may require AWS intervention.

free -m

1. **Identify memory-consuming processes:**

top

# Press 'Shift + M' to sort by memory usage

1. **Restart memory-intensive services:**

sudo systemctl restart <service-name>

1. **Add swap space** as a temporary solution:

sudo fallocate -l 2G /swapfile

sudo chmod 600 /swapfile

sudo mkswap /swapfile

sudo swapon /swapfile

1. **Optimize applications** for memory usage.
2. **Consider increasing instance memory** by changing the instance type.

**6.3 Disk I/O Performance Issues**

**Symptoms**

* Slow read/write operations
* High DiskReadOps or DiskWriteOps metrics
* Application performance degradation

**Steps to Resolve**

1. **Check available disk space:**

df -h

1. **Identify disk-intensive processes:**

iostat -dx

1. **Clear unnecessary files** to free up disk space:

# Remove old logs or temporary files

sudo rm -rf /var/log/\*.gz

1. **Optimize applications** to reduce disk I/O operations.
2. **Upgrade EBS volumes** to higher performance types if necessary.

ip addr show eth0

1. **Test network connectivity:**

ping <destination>

traceroute <destination>

1. **Check security group and network ACL configurations** to ensure proper traffic flow.
2. **Restart the network service:**

sudo systemctl restart network

# Ubuntu 24

sudo systemctl restart networking

1. **Review AWS VPC settings** if issues persist.

sudo systemctl status amazon-cloudwatch-agent

1. **Review the agent logs for errors:**

sudo tail -n 100 /opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log

1. **Restart the CloudWatch Agent:**

sudo systemctl restart amazon-cloudwatch-agent

1. **Verify the agent configuration file**, ensuring it's correctly defined:

sudo nano /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json

1. **Ensure the instance IAM role has the necessary permissions** to publish metrics to CloudWatch.
2. **Reinstall or update the CloudWatch Agent** if necessary.
3. **Configuration File**

{

"metrics": {

"metrics\_collected": {

"mem": {

"measurement": ["mem\_used\_percent"]

},

"disk": {

"measurement": ["disk\_used\_percent"],

"resources": ["/"]

}

}

}

}

# 7. Best Practices

# 7.1 Setting Realistic Thresholds

* Use performance baselines to set thresholds that reflect normal operating conditions.
* Avoid setting thresholds too low, which can result in false positives.

7.2 Reducing False Positives

* Implement sufficient evaluation periods before triggering alarms.
* Combine multiple metrics where appropriate (e.g., CPU and memory) to trigger alarms.

7.3 Regularly Reviewing Metrics and Alarms

* Periodically review CloudWatch metrics and adjust thresholds based on trends.
* Update alarms to reflect changes in application workload or instance configurations.

7.4 Automating Responses

* Use AWS Lambda functions or AWS Systems Manager Automation documents to automate remediation steps.
* Configure Auto Scaling policies for dynamic resource management.

# 8. AWS Systems Manager (SSM) Agent Troubleshooting

## 8.1 SSM Agent Status Commands

# Check SSM Agent status

sudo systemctl status amazon-ssm-agent

# View SSM Agent version

sudo amazon-ssm-agent -version

# Restart SSM Agent

sudo systemctl restart amazon-ssm-agent

# Enable SSM Agent on boot

sudo systemctl enable amazon-ssm-agent

## 8.2 Log Locations

OS-Specific Paths:

Amazon Linux/RHEL:

/var/log/amazon/ssm/amazon-ssm-agent.log

/var/log/amazon/ssm/errors.log

Ubuntu:

/var/snap/amazon-ssm-agent/current/amazon-ssm-agent.log

/var/snap/amazon-ssm-agent/current/errors.log

## 8.3 Installation/Reinstallation

Amazon Linux:

sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux\_amd64/amazon-ssm-agent.rpm

sudo systemctl status amazon-ssm-agent

RHEL 9:

sudo dnf install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux\_amd64/amazon-ssm-agent.rpm

sudo systemctl start amazon-ssm-agent

sudo systemctl enable amazon-ssm-agent

Ubuntu 24:

sudo snap install amazon-ssm-agent --classic

## 8.4 Common Issues and Solutions

Agent Not Connected

# Check connectivity

aws ssm describe-instance-information --region YOUR\_REGION

# Verify IAM role

aws iam get-instance-profile --instance-profile-name YOUR\_PROFILE

# Check proxy settings

cat /etc/systemd/system/amazon-ssm-agent.service.d/override.conf

Permission Issues

# Check SSM Agent permissions

sudo ls -l /var/lib/amazon/ssm/

sudo ls -l /var/log/amazon/ssm/

# Fix permissions

sudo chown -R root:root /var/lib/amazon/ssm/

sudo chmod -R 700 /var/lib/amazon/ssm/

Registration Issues

# Remove registration and restart

sudo rm -rf /var/lib/amazon/ssm/registration

sudo systemctl restart amazon-ssm-agent

## 8.5 Advanced Troubleshooting

Enable Debug Logging:

# Edit see log config

sudo vim /etc/amazon/ssm/seelog.xml

# Change to debug level

debug

# Restart agent

sudo systemctl restart amazon-ssm-agent

Network Connectivity Test:

# Test endpoint connectivity

nc -zv ssm.YOUR\_REGION.amazonaws.com 443

curl -v https://ssm.YOUR\_REGION.amazonaws.com

## 8.6 SSM Run Command Troubleshooting

Check Command Status:

# View command history

aws ssm list-commands --region YOUR\_REGION

# Get command details

aws ssm get-command-invocation \

--command-id "command-id" \

--instance-id "instance-id" \

--region YOUR\_REGION

## 8.7 Maintenance Tasks

Regular Updates:

# Update SSM Agent (Amazon Linux/RHEL)

sudo yum update amazon-ssm-agent -y

# Update SSM Agent (Ubuntu)

sudo snap refresh amazon-ssm-agent

## 8.8 Best Practices

Regular Monitoring:

Monitor agent status daily

Review error logs weekly

Keep agent updated

Security:

Use minimum required IAM permissions

Regular audit of SSM documents

Monitor session activity

Performance:

Clean old logs regularly

Monitor resource usage

Optimize command timeout values

## 9. References

* **Amazon EC2 Monitoring with CloudWatch**

[Viewing Amazon EC2 Metrics in Amazon CloudWatch](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/viewing_metrics_with_cloudwatch.html)

* **EC2 Status Checks**

[Monitoring Your Instances Using Status Checks](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html)

* **Installing and Configuring the CloudWatch Agent**

[Installing the CloudWatch Agent](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Install-CloudWatch-Agent.html)

* **AWS Observability Best Practices**

[AWS Observability Guide for EC2](https://aws-observability.github.io/observability-best-practices/guides/ec2/ec2-monitoring/)

* **CloudWatch Agent Configuration Reference**

[CloudWatch Agent Configuration File](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Agent-Configuration-File-Details.html)