SAT WEBSITE

AWS SSM Deployment Documentation

Generated: January 17, 2026

Version: 1.0

Project: sat-website

Owner: PARESHRANJAN299

Table of Contents

1. Executive Summary

This document provides comprehensive documentation for the SAT Website deployment infrastructure, which has been successfully migrated from SSH-based deployment to AWS Systems Manager (SSM) based deployment.

Aspect	Details
Project	SAT Website (sat-website)
Deployment Method	AWS SSM via GitHub Actions
Security Level	Enterprise-grade (No SSH keys)
Automation	Fully automated CI/CD pipeline
Deployment Time	~16 seconds (zero downtime)
Owner Auto-Deploy	Yes (no approval required)
Team Deploy	Requires owner approval
Region	us-east-1 (N. Virginia)

Key Achievements:

- Eliminated SSH key management and security risks
- Implemented IAM-based authentication with least privilege access
- Automated deployment with GitHub Actions (16-second deployments)
- Achieved zero-downtime deployments with Gunicorn reload
- Established audit trail for all deployments
- Configured environment-based access control

2. Architecture Overview

2.1 Infrastructure Components

Component	Service	Purpose
Source Control	GitHub	Code repository & CI/CD trigger
CI/CD	GitHub Actions	Automated deployment workflow
Authentication	AWS IAM	Identity & access management
Deployment	AWS SSM	Secure command execution
Web Server	EC2 (t3.micro)	Application hosting
App Server	Gunicorn	WSGI HTTP server
Framework	Flask	Python web framework
Reverse Proxy	Nginx	HTTP/HTTPS traffic routing

2.2 Data Flow

1. Developer pushes code to GitHub main branch

2. GitHub Actions workflow triggers automatically

3. Workflow authenticates with AWS using IAM credentials

4. GitHub Actions sends SSM command to EC2 instance

5. SSM agent on EC2 receives and executes commands as deploy user

6. Git pulls latest code, updates dependencies

7. Gunicorn service reloads (zero downtime)

8. Website updates at https://www.sat.net.in/

3. Security Configuration

3.1 Security Principles

Security Layer	Implementation	Status
Authentication	AWS IAM (no SSH keys)	✓ Active
Authorization	Least privilege IAM policies	✓ Active
Encryption	TLS 1.2+ (HTTPS)	✓ Active
Access Control	Environment-based approvals	✓ Active
Audit Trail	GitHub Actions logs	✓ Active
Network Security	VPC, Security Groups	✓ Active
Secrets Management	GitHub Secrets (encrypted)	✓ Active

3.2 Eliminated Security Risks

- **SSH Key Exposure:** No SSH keys stored in GitHub or distributed to team members

- **Static IP Dependency:** No need to whitelist IPs or manage network access

- **Credential Rotation:** IAM credentials can be rotated without workflow changes

- **Unauthorized Access:** GitHub Actions environment protection prevents unauthorized deployments

- **Man-in-the-Middle:** All communication over AWS's encrypted channels

4. IAM Policies & Permissions

4.1 IAM User: github-actions-sat

Policy Name	Type	Purpose
AmazonEC2ReadOnlyAccess	AWS Managed	Read EC2 instance information
AmazonSSMFullAccess	AWS Managed	Full SSM operations
GitHubActionsSSMDeploy	Custom	Minimal deployment permissions

4.2 Custom Policy: GitHubActionsSSMDeploy

This custom policy provides minimal permissions required for deployment:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:SendCommand",
                "ssm:GetCommandInvocation",
                "ssm:ListCommandInvocations",
                "ssm:DescribeInstanceInformation"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["ec2:DescribeInstances"],
            "Resource": "*"
        }
    ]
}
```

4.3 EC2 IAM Role

Role	Policy	Purpose
EC2-SSM-Role	AmazonSSMManagedInstanceCore	Allow SSM agent to communicate with AWS

5. Deployment Workflow

5.1 Workflow Triggers

Trigger	Action	Auto-Deploy
Owner pushes to main	Deploy immediately	Yes (production-auto)
Team pushes to main	Wait for approval	No (requires owner approval)
Manual trigger	Via workflow_dispatch	Based on actor

5.2 Deployment Steps

Step 1: Check commit author (owner vs. team member)

Step 2: Select environment (production-auto vs. production)

Step 3: Checkout code from GitHub

Step 4: Configure AWS credentials using IAM access keys

Step 5: Send SSM command to EC2 instance

Step 6: Execute deployment commands as deploy user

Step 7: Wait for command completion

Step 8: Retrieve and display deployment output

5.3 Deployment Commands Executed on EC2

1. Switch to deploy user: `sudo -u deploy`

2. Navigate to project: `cd /var/www/sat/sat-website`

3. Fetch latest code: `git fetch origin main`

4. Reset to latest: `git reset --hard origin/main`

5. Activate virtualenv: `source venv/bin/activate`

6. Update pip: `pip install --upgrade pip`

7. Install dependencies: `pip install -r requirements.txt`

8. Reload service: `sudo systemctl reload sat`

6. GitHub Actions Configuration

6.1 GitHub Secrets

Secret Name	Environment	Description
AWS_ACCESS_KEY_ID	Both	IAM user access key
AWS_SECRET_ACCESS_KEY	Both	IAM user secret key
AWS_REGION	Both	AWS region (us-east-1)
EC2_INSTANCE_ID	Both	EC2 instance ID (i-03a82e4a84a456ef4)

6.2 GitHub Environments

Environment	Required Reviewers	Used By	Purpose
production-auto	None	Owner (PARESHRANJAN299)	Auto-deploy without approval
production	Owner	Team members	Deploy with owner approval

6.3 Branch Protection Rules

- **Repository admin bypass:** Enabled (owner can push directly to main)
- **Status checks:** Required (GitHub Actions must pass)
- **Branch up to date:** Required before merging
- **Include administrators:** Disabled (allows owner bypass)

7. EC2 Configuration

7.1 Instance Details

Property	Value
Instance ID	i-03a82e4a84a456ef4
Instance Type	t3.micro
Name	SAT_EC2_WebServer
Region	us-east-1 (N. Virginia)
Availability Zone	us-east-1c
OS	Ubuntu 24.04 LTS
Public IP	34.228.165.210
Private IP	172.31.22.111

7.2 User Accounts

User	UID	Groups	Purpose
deploy	1001	deploy, sudo	Deployment & application owner
paresh	1002	paresh, sudo, deploy	System administration

7.3 Directory Structure

- `/var/www/sat/sat-website/` - Main project directory
- `/var/www/sat/sat-website/venv/` - Python virtual environment
- `/var/www/sat/sat-website/app.py` - Flask application
- `/var/www/sat/sat-website/templates/` - HTML templates
- `/var/www/sat/sat-website/static/` - Static assets (CSS, JS, images)

7.4 Sudoers Configuration

File: `/etc/sudoers.d/deploy`

```
deploy ALL=(ALL) NOPASSWD: /bin/systemctl reload sat
deploy ALL=(ALL) NOPASSWD: /bin/systemctl restart sat
deploy ALL=(ALL) NOPASSWD: /bin/systemctl status sat
```

7.5 Systemd Service

Service: `sat.service`

- **Location:** /etc/systemd/system/sat.service
- **Status:** Active (running)
- **Workers:** 3 Gunicorn workers
- **Bind:** 127.0.0.1:8000 (proxied via Nginx)
- **User:** deploy
- **Working Directory:** /var/www/sat/sat-website

8. Deployment Flow Diagram

Step	Component	Action	Result
1	Developer	Push to main branch	Code committed
2	GitHub	Trigger workflow	Actions start
3	GitHub Actions	Check commit author	Determine environment
4	GitHub Actions	Load secrets	AWS credentials ready
5	GitHub Actions	Configure AWS CLI	Authenticated with AWS
6	GitHub Actions	Send SSM command	Command transmitted
7	AWS SSM	Route to EC2 instance	Command received
8	SSM Agent	Execute as deploy user	Commands run
9	Deploy User	Git fetch & reset	Code updated
10	Deploy User	Install dependencies	Packages updated
11	Deploy User	Reload Gunicorn	Service reloaded
12	Gunicorn	Zero-downtime reload	New code active
13	GitHub Actions	Retrieve output	Deployment confirmed
14	Website	Serve new version	✓ Live

9. Troubleshooting Guide

9.1 Common Issues & Solutions

Issue	Cause	Solution
Exit code 255	SSM command failed	Check SSM agent status, verify IAM permissions
Permission denied	User cannot access files	Verify directory ownership: chown -R deploy:deploy
Service reload fails	Sudoers not configured	Create /etc/sudoers.d/deploy file
AWS credentials invalid	Expired or incorrect keys	Regenerate IAM access keys in AWS Console
Workflow syntax error	Invalid YAML	Validate YAML syntax online
Instance not found	Wrong instance ID	Verify EC2_INSTANCE_ID in GitHub secrets

9.2 Verification Commands

- **Check SSM Agent:** `sudo systemctl status amazon-ssm-agent`

- **Test SSM Connection:** Use Session Manager in AWS Console

- **Verify deploy user:** `id deploy`

- **Check directory ownership:** `ls -la /var/www/sat/sat-website`

- **Test git access:** `sudo -u deploy git -C /var/www/sat/sat-website status`

- **Check service status:** `sudo systemctl status sat`

- **View recent logs:** `sudo journalctl -u sat -n 50`

10. Maintenance Checklist

10.1 Daily Checks

■ Monitor GitHub Actions for failed deployments

■ Check website availability: https://www.sat.net.in/

■ Review deployment logs for errors or warnings

10.2 Weekly Checks

■ Review EC2 CloudWatch metrics (CPU, memory, disk)

■ Check SSM agent status and version

■ Verify backup procedures are working

■ Review IAM user access logs in CloudTrail

10.3 Monthly Checks

■ Rotate IAM access keys (security best practice)

■ Update EC2 security patches: `sudo apt update && sudo apt upgrade`

■ Review and update Python dependencies

■ Test disaster recovery procedures

■ Review and update documentation

10.4 Quarterly Checks

■ Conduct security audit of IAM policies

■ Review GitHub Actions usage and costs

■ Update Python and system packages to latest LTS versions

■ Review and optimize EC2 instance type based on usage

■ Conduct penetration testing

Quick Reference Card

Resource	Value / Command
Website	`https://www.sat.net.in/`
GitHub Repo	`https://github.com/PARESHRANJAN299/sat-website`
GitHub Actions	`https://github.com/PARESHRANJAN299/sat-website/actions`
EC2 Instance ID	`i-03a82e4a84a456ef4`
AWS Region	`us-east-1`
IAM User	`github-actions-sat`
Deploy User	`deploy`
Project Path	`/var/www/sat/sat-website`
Service Name	`sat.service`
Emergency Commands	
Restart service	`sudo systemctl restart sat`
View logs	`sudo journalctl -u sat -n 100`
Check SSM agent	`sudo systemctl status amazon-ssm-agent`
Manual deploy	`cd /var/www/sat/sat-website && sudo -u deploy git pull`

Support Contacts

Owner: PARESHRANJAN299

Documentation Version: 1.0

Last Updated: January 17, 2026