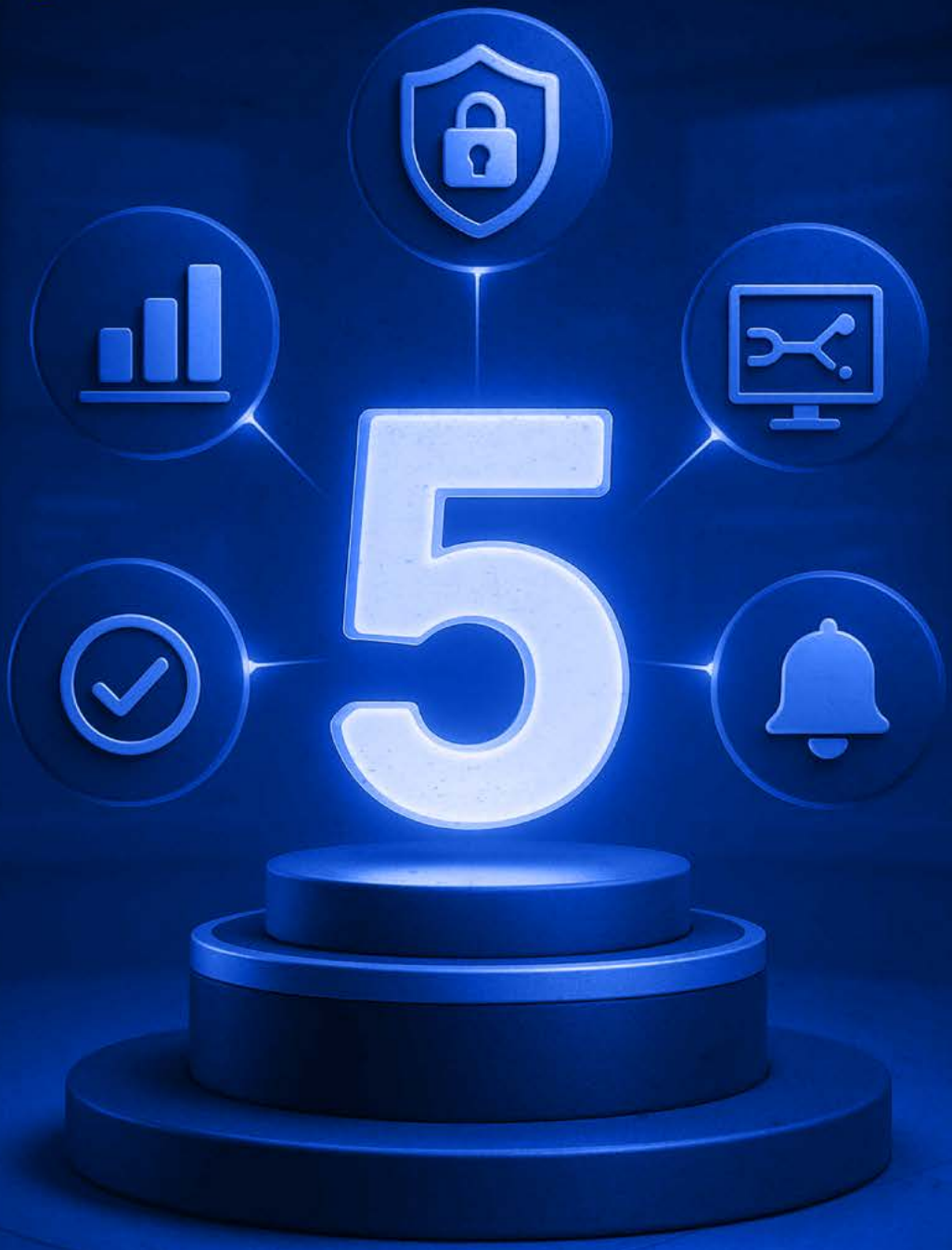
ManageEngine
Site24x7

5 Steps to Monitor and Secure Customer-Facing Applications in BFSI



What are we going to cover?

Introduction

Banking, financial services, and insurance (BFSI) have always operated under tight regulatory oversight. Every few years, new frameworks emerge (e.g., PSD2, GLBA, DORA) that set even higher standards for how companies should handle data, manage risks, and respond to incidents.

As BFSI organizations modernize their tech stacks by moving toward containerized architectures, hybrid cloud setups, and serverless apps, regulators are responding with even stricter requirements. These rules increasingly focus on operational resilience, customer data protection, and third-party risk.

At the same time, BFSI remains one of the top targets for cybercriminals. The sheer volume and sensitivity of the financial and personal data they handle make them irresistible to malicious actors seeking to exploit vulnerabilities for profit or sabotage.

All of this puts pressure on BFSI companies to not only meet compliance checklists but also rethink how they monitor and secure their customer-facing applications. These apps are often the front door to their services. Any downtime, breach, or data leak can directly damage customer trust, brand reputation, and business continuity.

This e-book goes through five key steps that BFSI companies can take today to monitor and secure their customer-facing applications for proactive defense and continuous operational integrity.

Step 01

Identify and Prioritize Customer-Facing Assets

Before you can secure or monitor anything, you need a clear picture of what needs protection. In BFSI, this starts with identifying every digital asset that interacts with customers. These are the entry points attackers often target and the places where customers experience your service directly. Missing just one can lead to blind spots in your monitoring and gaps in your security coverage.

Here's how to tackle this step in a structured way:

List All Public-Facing Digital Touchpoints

Start by documenting all your customer-facing services, applications, or interfaces, such as:

- Websites (corporate sites, self-service portals)
- Mobile apps (iOS, Android)
- Customer support platforms (live chat, support widgets)
- APIs (especially those used by fintech partners or aggregators)
- Chatbots and messaging interfaces
- Payment gateways
- Internet-banking or insurance self-service portals

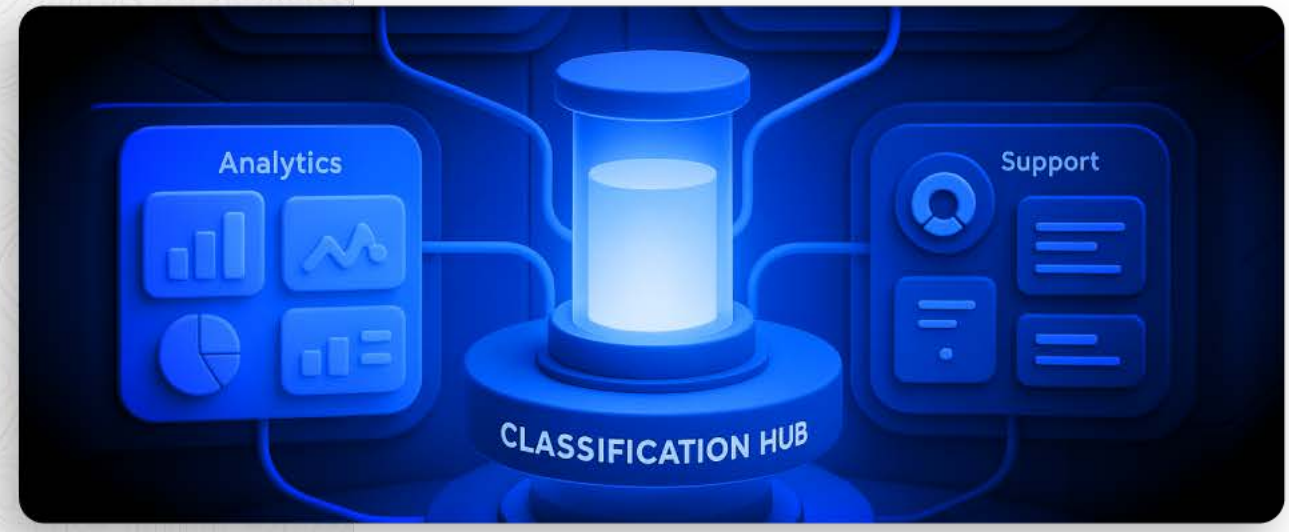
This list should cover both primary apps and any secondary systems that support or display customer-facing features.

Categorize by Business Function and Criticality

Once everything is listed, group assets based on their function and how critical they are to business operations or customer experience. For example, one way to categorize could be:

- **High criticality:** Login portals, transaction APIs, payment systems, onboarding flows
- **Medium criticality:** Account dashboards, chatbot interfaces, policy or claims submission portals
- **Low criticality:** Informational microsites, product brochures, career portals

This helps prioritize which assets need stronger monitoring and faster alerting.



Maintain a Real-Time Inventory

Leverage a tool that can help you maintain and visualize a real-time inventory. Here's why the real-time aspect matters in inventory:

- You'll know immediately if a new API goes live without monitoring
- You can detect shadow IT or misconfigured assets that slip through
- It allows teams to respond faster during an incident

Tag and Map Ownership

Each asset should be tagged with metadata like application owner, environment (production, staging), hosting provider, and business unit, etc. This makes it easier to assign responsibility when performance or security issues arise.

Site24x7 supports automatic tagging and grouping on the fly, which reduces manual effort and avoids errors. You can then organize and filter assets by environment, location, or criticality — all within your dashboards.

Step 02

Monitor Infrastructure Health and Uptime

Once your customer-facing assets are identified and prioritized, the next step is to make sure the infrastructure behind them is healthy and available at all times. This includes everything from cloud computing resources to DNS services. Even a small outage in one layer of the stack can impact hundreds of users, trigger compliance incidents, or create customer churn.

Here's how to monitor your infrastructure the right way:

Start with the Core Infrastructure

Monitor the foundational layers that power your applications, starting from backend servers, databases, and app servers, to load balancers, firewalls, and gateways, and finally to DNS and CDNs. If any of these fail or slow down, it can affect customer access, even if your main application is technically running.

Site24x7's wide coverage makes this easier by providing out-of-the-box plugins and checks for critical components like DNS servers, CDNs, and network devices.

- Databases: Measure query response times, spot connection issues, and detect replication lag.
- Cloud services: Get native integration and metrics for AWS, Azure, and Google Cloud resources.

...and the list doesn't end here. Other parts of your infrastructure that Site24x7 can monitor include caches, load balancers, CI/CD tools, big data platforms, collaboration apps, and security solutions, etc. Go through the full list of integrations/plugins here.

Collect and Analyze Logs and Events

Go beyond metrics to monitor logs and events. They will prove valuable during troubleshooting sessions and help spot issues like brute force attacks, bad deployments, or config errors before they turn into outages.

- Ingest logs from firewalls, databases, API gateways, web servers, etc.
- Use a SIEM or log aggregator (like Site24x7's log management)
- Set up pattern-based alerts (e.g., 5 failed logins in 1 minute, repeated 500 errors)

Site24x7's event log monitoring also helps detect suspicious activity in Windows and Linux environments.

Track Key Performance Metrics

Set up active and passive monitoring to track:

- Availability: Is the service reachable from multiple regions?
- Latency: How long do API calls, page loads, or transactions take?
- DNS resolution time: Time taken for domain name lookups.
- Error rates: HTTP 5xx/4xx responses, timeouts, dropped packets, or failed transactions.
- Network throughput: Volume of data transferred to from services.
- Resource usage: CPU, memory, disk I/O, and bandwidth across nodes.
- Database connection pools: Number of open/idle connections.
- Container orchestration health: Status of pods, nodes, and deployments.

Site24x7's dashboards make it easy to see all these metrics together, so you don't miss any early warning signs.

Define Alert Thresholds and Escalation Policies

Don't wait until customers complain. Configure thresholds that trigger alerts when metrics cross certain levels, such as CPU usage over 90% for 5 minutes, or API latency spikes above 1 second. With Site24x7's flexible alarms engine, you can set multi-condition thresholds and automated actions for remediation.

Set escalation rules for different severity levels:

- Critical: Notify on-call engineers immediately (SMS, PagerDuty, etc.)
- Warning: Log the issue and notify the team via email or Slack
- Info: Store in logs for audit and trend analysis

Time-to-response is everything in BFSI, where delays affect customer trust.

Use Tools That Cover Multiple Layers

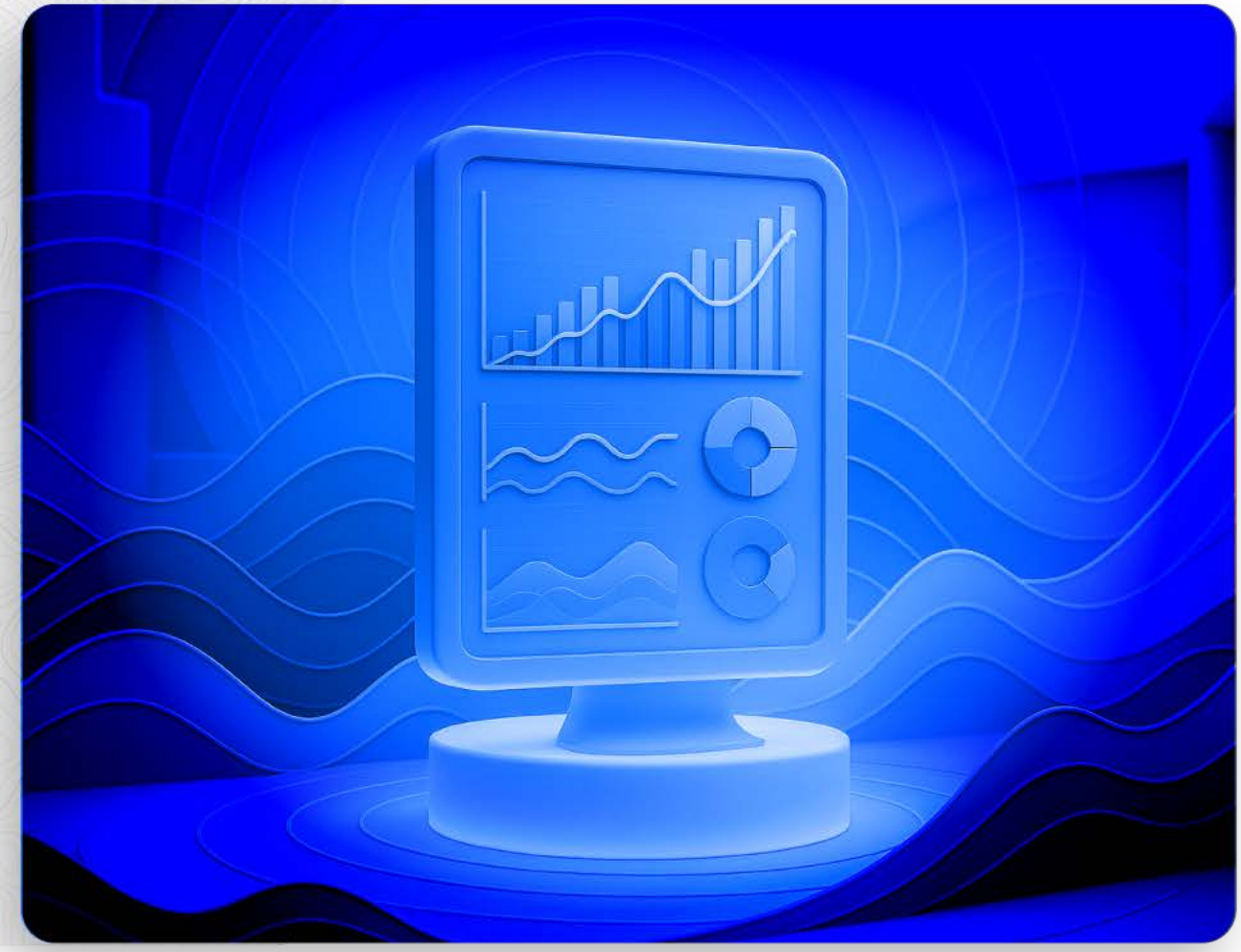
Invest in an all-in-one monitoring platform like Site24x7 to avoid tool sprawl and reduce operational overhead. With everything in one place, administrators don't need to switch between multiple tools to get a complete view of infrastructure health. Site24x7 covers:

- Servers: Monitor OS-level health, running processes, and available disk space.
- Containers: Track Kubernetes clusters, container resource usage, and pod status.

Track Historical Trends and Seasonal Patterns

Use Site24x7's reports to review historical performance trends and:

- Predict usage spikes (e.g., tax season, loan deadlines)
- Right-size infrastructure to avoid over- or under-provisioning
- Spot recurring incidents tied to deployments or updates



Step 03

Monitor Application Performance

At the end of the day, if your application is slow, buggy, or inconsistent, it won't matter if your infrastructure is healthy. Application performance monitoring (APM) helps you understand what's happening inside your app *— i.e., how each request behaves, where the slowdowns are, and what's affecting the customer experience.*

In BFSI, the stakes are considerably higher than elsewhere. A slow credit card application form, a failed login during peak hours, or a crash in the mobile app during a transaction can directly lead to lost business, customer complaints, and even regulatory attention.

Here's how you can implement robust application monitoring:

Use Application Performance Monitoring (APM) tools

APM tools help trace the full journey of a request: from the moment a user clicks a button to the point where the response is sent. This lets you:

- Identify slow functions or API calls
- Trace issues down to specific lines of code

- Monitor backend service-to-service communication latency
- Spot memory leaks or resource bottlenecks in real time

Site24x7's APM Insight gives you distributed tracing out-of-the-box. You can track requests as they flow across multiple services and APIs, which is great for pinpointing root causes in complex banking and insurance applications.

Monitor Real User Experience (RUM)

RUM collects data directly from real users as they interact with your web or mobile apps. It gives you insights into:

- Page load times by region, browser, or device
- UI responsiveness during peak hours
- Frontend errors like JavaScript crashes or broken elements

Site24x7's Real User Monitoring (RUM) module helps you see exactly how customers experience your apps and where performance dips are hurting your KPIs. For example, if RUM data shows that users in a certain region consistently see slower response times, you'll know where to optimize.

Don’t Forget Mobile RUM

Many BFSI customers now rely on mobile apps for daily banking, insurance claims, or investment tracking. Site24x7 supports mobile RUM so you can track:

- App launch times
- Screen rendering delays
- Crashes and freezes
- Backend call failures from mobile SDKs

This is especially important when users are on different devices, networks, or OS versions.

Track Third-Party Service Impact

Modern BFSI apps rely on many third-party services: payment gateways, credit scoring APIs, KYC platforms, or embedded chat systems. If any of these slow down or fail, it reflects badly on your app, and it won’t matter that the problem isn’t on your side.

Site24x7 lets you monitor external APIs and web services separately and set thresholds for timeouts or failures so you can:

- Isolate issues quickly
- Build fallback workflows
- Escalate to vendors before customers notice

Use Synthetic Monitoring for Predictability

Synthetic monitoring simulates real user interactions at regular intervals, such as logging in, checking account balances, or submitting an insurance claim. Use the free Synthetic Monitoring Tool in Site24x7 to:

- Test from different regions
- Catch issues outside business hours
- Monitor APIs and flows that aren't used frequently

It's especially useful before new releases or during maintenance windows to make sure everything works as expected.

Tie Metrics to Business Outcomes

Site24x7's integrated dashboards help you correlate app performance metrics with user impact, so technical teams can prioritize fixes that make the biggest difference for your customers and your bottom line. For example:

- Slow page loads → Increased bounce rate
- Failed login attempts → Abandoned sessions or customer support calls
- Cart or form drop-offs → Lost applications or sales
- API latency spikes → Delayed transactions

Step 04

Track User Behavior and Operational Metrics

Even when your infrastructure and application seem healthy, your users could still be struggling. A broken login flow, a form that won't submit, or a page that silently fails to load can all go unnoticed… unless you're actively tracking how users behave inside the application.

User behavior data, when combined with performance and operational metrics, gives you the full picture. It helps detect hidden issues, prioritize improvements, and understand the real-world impact of technical problems on customer experience.

Here's how to do it effectively:

Monitor Key User Flows

Start by tracking core user interactions; especially those tied to business-critical goals:

- Login success/failure rates: Spikes in failed logins may be because of authentication issues, backend outages, or brute-force attempts.
- Session times: Sudden drops may suggest frustration or app crashes.
- Bounce rates: High bounce rates from key entry points (e.g. onboarding, loan application) could point to broken features or poor performance.

Site24x7's Real User Monitoring (RUM) helps you collect this behavioral data directly from real users and spot where things are breaking down. It gives visibility into frontend performance issues that often go undetected when you focus solely on backend monitoring.

Detect Behavioral Anomalies

Use the RUM dashboard in Site24x7 to spot any sudden or unusual changes in user behavior. For example:

- A drop in completed transactions
- Users repeatedly refreshing a page
- Increased clicks on error-prone elements
- Traffic suddenly falling from a specific region or device type

These may point to outages, misconfigurations, or even security problems (e.g., traffic being blocked or rerouted unexpectedly).

Analyze Usage Trends to Predict Problems

Use RUM reports to track usage patterns over time. This should help you spot:

- **Capacity issues:** If concurrent sessions or data usage grows faster than expected, you may need to scale infrastructure.
- **Misconfigurations:** Users avoiding certain features or flows could mean something's broken, poorly designed, or too slow.
- **Adoption gaps:** These are features that were released but show very little engagement.

Use these trends to inform both technical decisions (like resource allocation) and business ones (like feature rollbacks or redesigns).

Combine Behavioral Data with Technical Metrics

A failed login isn't just a failed login; it could be due to:

- A downed API
- Slow database response
- A misconfigured firewall
- A frontend bug on mobile devices

This is why it's important to combine behavioral insights with APM, infrastructure monitoring, and error logs — something Site24x7 offers out-of-the-box. It will help you pinpoint the real root cause faster, and fix what actually matters to users.

Set Thresholds and Automate Reporting

Establish baselines for behavior, such as expected session length, form submission rates, or login success percentages. Trigger alerts when metrics drop below acceptable levels.

Site24x7 allows you to define alert conditions across RUM and APM metrics and set up automated weekly/monthly reports that surface behavioral anomalies and user-impacting issues for both technical and business stakeholders.

Step 05

Strengthen Security Hygiene

By now, your monitoring setup should give you visibility into infrastructure, applications, and user behavior. The final step is to tighten your basic security hygiene. Here are some tips to do that:

- Use Site24x7's Log Management to collect and analyze security-related logs from firewalls, authentication systems, and application servers. This helps you catch suspicious activity, failed logins, brute-force attempts, or unusual access patterns early on.
- Multi-factor authentication should be turned on for all admin-level access points, including cloud consoles, internal dashboards, CI/CD tools, and remote access systems. This applies to both employees and third-party vendors.
- Keep a close watch on login activity across your systems. Track failed login attempts, unusual geographic access patterns, logins outside business hours, or multiple sessions from different locations. Any irregular behavior should trigger alerts and be tied into your broader incident response plan to ensure early action.
- Separate your environments such that public-facing services don't have direct access to internal databases or admin systems. Set up internal firewalls or use cloud-native controls like security groups or VPC rules to enforce strict communication policies between services. Default to deny, and allow only what's explicitly needed.

- Give users the minimum access they need to do their job. This applies to internal staff, service accounts, and third-party integrations. Site24x7 offers role-based access control features to support this.
- Run simulated breach scenarios and measure how quickly teams identify and respond to alerts. Review escalation paths, test runbooks, and make sure responsibilities are clear during an incident.
- Avoid hardcoding credentials in your codebase or storing them in unsecured locations. Use a proper secrets management tool to store and rotate secrets. Track who accessed what, and make sure audit logs are enabled.

Best Practices for Implementation

Finally, here are some best practices that will help you avoid common challenges during implementation:

- Assign clear ownership for each monitoring domain: infra, application, user behavior, and security.
- Create a centralized configuration file or repository for monitoring rules, thresholds, and alert logic to ensure consistency across teams.
- Integrate monitoring setup into infrastructure-as-code and deployment pipelines so nothing goes live without visibility.
- Define tagging standards for resources (e.g. by environment, criticality, owner) to make automated grouping and filtering possible.
- Run test alerts regularly in staging environments to confirm alerting logic, thresholds, and escalation paths are working as expected.
- Schedule regular review sessions between dev, security, and ops teams to align monitoring scope with changes in the tech stack.
- Document monitoring coverage per application or service as part of internal onboarding and handover processes.
- Build a checklist for provisioning new services that includes monitoring, alerting, and logging steps before go-live.
- Automate the cleanup of unused monitors and stale alerts to keep the monitoring setup lean and manageable.

- Use templates and repeatable playbooks when rolling out monitoring for new environments or regions to save time and reduce errors.
- Include monitoring and security validation as part of your code review or pull request checklist to catch gaps early.
- Standardize log formats and retention policies across services to simplify correlation and forensic analysis during investigations.
- Use canary deployments and monitor their metrics closely before rolling out changes to the wider customer base.
- Use role-based access control in your monitoring and logging tools to prevent unauthorized access to sensitive data or configurations.

Conclusion

To meet changing regulatory requirements and future-proof your infrastructure, it's important to take a systematic approach to monitoring and securing customer-facing BFSI apps. We hope this guide gave you a clear, step-by-step path to get started.

About ManageEngine Site24x7

ManageEngine Site24x7 enables IT and DevOps teams to monitor server environments seamlessly with real-time visibility into performance metrics such as CPU, memory, disk, and network utilization. Its unified platform correlates metrics, logs, and events to provide a single view of server health across hybrid and cloud infrastructures. With intelligent alerting, automated remediation, and prebuilt dashboards, Site24x7's server monitoring helps teams detect anomalies early, troubleshoot efficiently, and maintain optimal performance across dynamic workloads.

Get Quote | Request Demo