



# Maximize the power of Amazon Bedrock deployments

with Dynatrace  
AI-powered  
observability

# Amazon Bedrock

## MODELS AND INFRASTRUCTURE

- Models
- Evaluations
- Inference optimization

## CUSTOMIZATION

- Fine tuning
- Knowledge bases
- Model distillation

## GUARDRAILS

- Responsible AI
- Data protection
- Governance
- Hallucination controls

## AGENTIC AI

- Runtime
- Memory
- Identity
- Gateway
- Code Interpreter
- Browser
- Observability
- Policy
- Evaluations

Built-in security

## INTRODUCTION

# What is Amazon Bedrock?

Amazon Bedrock is a fully managed cloud service from Amazon Web Services designed to help engineering teams easily build and scale generative AI applications.

It provides access to a range of foundational large language models (LLMs) and other generative AI models from leading providers, including Amazon, Anthropic, Cohere, Stability AI, and AI21 Labs.

With Amazon Bedrock, developers can experiment with, customize, and deploy generative AI models for tasks such as text generation, image creation, code synthesis, and more – all without managing underlying infrastructure.

# Why engineering teams value Amazon Bedrock and its key features

---

Key features of Amazon Bedrock include the following:

- **Model variety.** Access multiple high-quality, pre-trained models from top AI companies, allowing users to choose the best fit for their use case.
- **Customization.** Amazon Bedrock enables organizations to fine-tune models with their own data, ensuring outputs are tailored to specific business needs.
- **Serverless deployment.** As a managed service, Amazon Bedrock abstracts infrastructure management, allowing developers to focus on building applications rather than provisioning hardware.
- **Secure and scalable.** Built on Amazon's secure cloud platform, Amazon Bedrock offers enterprise-grade security and the ability to scale workloads as needed.
- **Amazon Ecosystem integration.** Amazon Bedrock integrates seamlessly with other AWS services, such as Amazon S3, AWS Lambda, and Amazon SageMaker, enabling robust, end-to-end AI solutions.

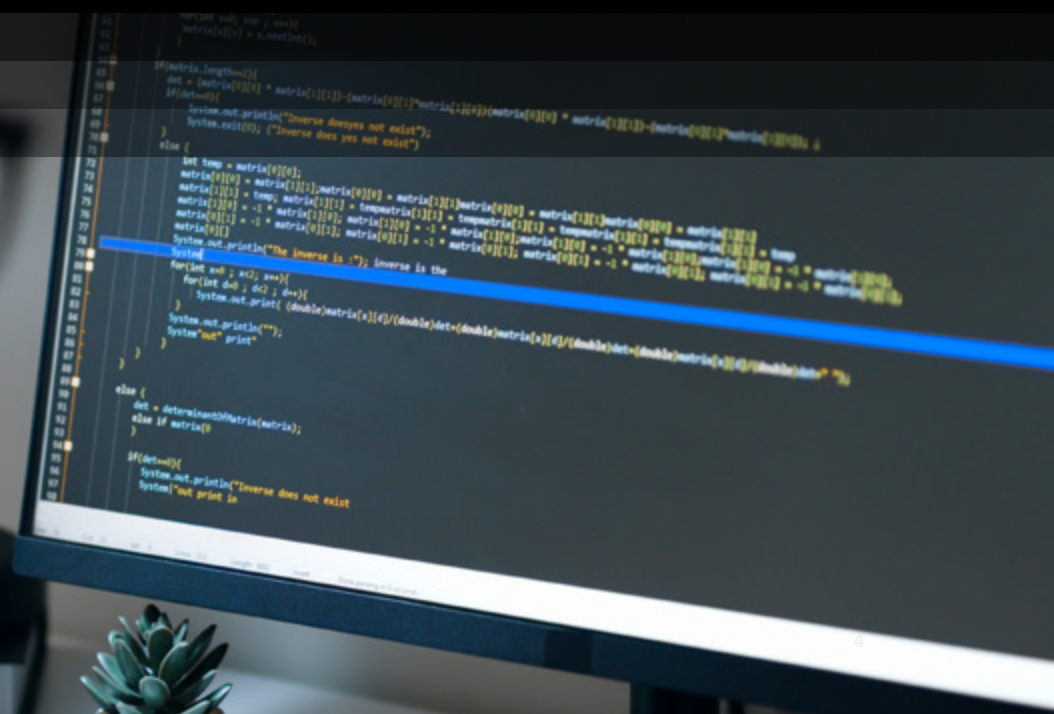
Many engineering teams turn to Amazon Bedrock as they look to integrate generative AI into their workflows and applications. By providing seamless access to a diverse set of advanced AI models, Amazon Bedrock accelerates innovation and shortens development cycles, making it easier to launch new features and products quickly.

Amazon Bedrock also simplifies the deployment of AI technologies. By leveraging a serverless architecture, the technology lowers barriers to entry, enabling companies without deep artificial intelligence expertise to build and deploy robust AI solutions while only paying for what they use.

Additionally, its integration with Amazon's secure cloud infrastructure ensures enterprise-grade security, compliance, and scalable customization, giving organizations confidence in both operational efficiency and data protection.

This flexible architecture enables enterprises to develop AI-powered applications at a significantly faster pace. However, the proper guardrails need to be put in place to ensure AI-enabled applications operate securely and effectively.

When deploying an application or agent to Amazon Bedrock, monitoring key performance indicators (KPIs) is essential to ensure optimal operation, efficiency, and security.



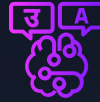
```
61 int main() {
62     int a, b, c, d;
63     cout << "Enter elements of a 2x2 matrix: ";
64     for(int i=0; i<2; i++)
65         for(int j=0; j<2; j++)
66             cin >> matrix[i][j];
67     if(matrix[0][0] == 0)
68         cout << "Inverse does not exist" << endl;
69     else {
70         int temp = matrix[0][0];
71         matrix[0][0] = matrix[1][1];
72         matrix[1][1] = temp;
73         matrix[0][1] = -1 * matrix[0][1];
74         matrix[1][0] = -1 * matrix[1][0];
75         matrix[0][1] = temp * matrix[1][0];
76         matrix[1][0] = temp * matrix[0][1];
77         System.out.println("The inverse is:");
78         System.out.println("Inverse is the");
79         for(int i=0; i<2; i++)
80             for(int j=0; j<2; j++)
81                 System.out.print((double)matrix[i][j]/(double)det << " ");
82         System.out.println("");
83         System.out.println("");
84         System.out.print("");
85     }
86 }
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

# Critical key performance indicators



## INFRASTRUCTURE AND APM

- Model latency
- Throughput
- Resource utilization
- Technical error rate



## MODEL AND LLM OBSERVABILITY

- Token consumption and cost
- Security and compliance guardrails
- Model accuracy and quality
- Model version comparison

## SECTION TWO

# Critical KPIs to maximize the value of Amazon Bedrock deployments

---

To maximize the value delivered by Amazon Bedrock deployments, organizations must move beyond a one-size-fits-all monitoring approach. True visibility requires distinguishing between the foundational performance of your application infrastructure and the unique, dynamic behaviors of generative AI models.

By tracking these distinct categories of KPIs, you can maintain high standards of service, optimize resource allocation, and innovate with confidence.

## **Infrastructure and application performance (APM)**

Ensure your underlying systems are resilient and scalable by monitoring these core operational metrics:

### **1. MODEL LATENCY**

Track the response time of generative AI models to user queries. Low latency is crucial for delivering a smooth, responsive user experience, particularly in real-time or customer-facing applications like service agents.

### **2. THROUGHPUT**

Measure the number of requests processed per second. High throughput indicates your system's ability to handle increased usage or traffic spikes without performance degradation, ensuring reliability at scale.

### **3. RESOURCE UTILIZATION**

Monitor critical infrastructure metrics, including CPU, memory, and network usage. Visibility into these resources helps optimize costs, prevent bottlenecks, and ensure your applications scale efficiently across your cloud environment.

### **4. TECHNICAL ERROR RATE**

Observe the percentage of failed requests or system operations. A rising error rate often signals integration issues or system instability that requires immediate attention to prevent downtime.

## **Model and LLM observability**

Gain insight into the cognitive behavior, cost, and safety of AI deployments with these specialized metrics:

### **1. TOKEN CONSUMPTION AND COST**

Track spending on compute, storage, and API usage, specifically focusing on token consumption. Monitoring these cost-related KPIs allows for precise budget control and informs strategic decisions regarding scaling and optimization.

### **2. SECURITY AND COMPLIANCE GUARDRAILS**

Monitor guardrails to ensure safety, privacy, and compliance. Acting as a critical layer of control, these metrics track how effectively a system intercepts and filters user inputs and model outputs to block sensitive or inappropriate content.

### **3. MODEL ACCURACY AND QUALITY**

Assess the relevance of AI outputs against business objectives and trustworthy data sources. Measuring the “grounding” of a model ensures outputs are explainable and derived from corporate data sources rather than hallucinations, building trust in AI agents.

### **4. MODEL VERSION COMPARISON**

With new models and versions releasing frequently, it is vital to ensure the application utilizes the most effective option. Leverage practices like A/B testing to compare models in real time, ensuring continuous improvement and adaptation.

# How to operationalize AI observability

---

## Unified observability

Many organizations rely on different solutions when developing AI applications or agents, rather than using the same observability solution that will be used in production. This approach can lead to inconsistent monitoring, gaps in visibility, and challenges in identifying and resolving issues that may only surface under real-world workloads.

Without unified observability from development through production, teams may miss critical metrics, encounter difficulty in correlating data across environments, and experience delays in troubleshooting or optimizing system performance.

Ultimately, this disconnect can undermine the reliability, security, and efficiency of AI deployments, making it harder to ensure the application meets business objectives and compliance standards.

## Standardized frameworks

**OpenTelemetry** and **OpenLLMetry** are invaluable tools for instrumenting AI applications and agents, providing standardized frameworks for collecting, analyzing, and correlating critical observability data.

By leveraging these solutions, teams can ensure consistent monitoring from development through production, enabling comprehensive visibility into model performance, user interactions, and system health. This unified observability helps quickly identify issues, optimize resource usage, and maintain compliance, supporting reliable and effective AI deployments.

## End-to-end traceability

End-to-end traceability is crucial for AI and agentic applications because it enables teams to track the entire lifecycle of data, decisions, and interactions within the API, frontend stack, and backend applications. This comprehensive visibility ensures that performance bottlenecks, errors, and anomalies can be quickly identified and addressed, minimizing downtime and maintaining high service quality.

By correlating **traces, logs, events, and metrics** across all components, organizations can ensure their AI applications function as intended, meet business objectives, and remain compliant with regulatory standards. Effective traceability also supports faster troubleshooting, enhances accountability, and provides the insights needed to continuously optimize both the model and underlying infrastructure.

## Feedback loops and proactive alerting

Feedback loops, alerting, and human-in-the-loop automation are essential components when operating agentic-based applications. Robust feedback loops enable continuous learning and adaptation by providing actionable insights into model performance and user interactions.

**Integrated alerting systems** facilitate rapid detection of anomalies or degradations in service, allowing teams to proactively address issues before they affect end users. Incorporating human-in-the-loop automation further enhances reliability and accountability, as it ensures that critical decisions or edge cases receive expert review, mitigating risks and supporting trustworthy AI outcomes.

## Unified view and visibility

Visualizing **model behavior, infrastructure health, user experience, performance, and cost trends** in a single unified view is critical when observing agentic applications. Centralized visualization streamlines correlation across signals, making it easier to detect anomalies, trace root causes, and understand the interplay between different system components.

When these key observability facets are distributed across disparate tools, teams face fragmented visibility, increased context-switching, and run the risk of missing important cross-domain insights. This disconnect can delay troubleshooting and lead to inconsistent data interpretation, jeopardizing reliability, cost-efficiency, and user satisfaction.

By consolidating all relevant observability data, organizations ensure a holistic, actionable view that supports faster decision-making and more resilient AI operations.



### **END-TO-END OBSERVABILITY**

across the entire  
AI infrastructure



### **REAL-TIME INSIGHT**

into model behavior  
and performance



### **CAUSAL AI**

to surface root cause



### **UNIFIED DASHBOARDS**

for observability  
and accountability

## SECTION FOUR

# The power of AI observability with Dynatrace

Dynatrace provides robust observability for agentic-based applications by monitoring every layer of the technology stack – from infrastructure and cloud resources to AI models and user interactions. By correlating data from metrics, traces, logs, and events in real time, Dynatrace empowers teams to visualize system health, performance, and behavior in a unified dashboard.

This end-to-end visibility enables faster root-cause analysis, proactive anomaly detection, and seamless troubleshooting, while AI-powered analytics drive smarter automation and alerting. As a result, organizations can optimize resource utilization, enhance reliability, ensure compliance, and deliver consistent, high-quality user experiences across their entire AI ecosystem.

## **End-to-end observability of AI-powered stacks**

Dynatrace captures telemetry across the entire AI infrastructure, extending from foundational cloud resources to granular, model-level signals such as latency, versioning, cost metrics, and detailed input/output traces.

By instrumenting every layer of the technology stack, Dynatrace seamlessly ingests and correlates data from infrastructure components, application services, and AI models, providing real-time insights into operational health and performance.

This unified approach enables teams to monitor individual model performance – including tracking version changes, latency spikes, resource consumption, and the flow of inputs and outputs – ensuring accurate root-cause analysis, cost management, and optimized user experiences throughout the AI lifecycle.

Dynatrace supports both self-hosted deployments and API-based services such as Amazon Bedrock. For self-hosted environments, Dynatrace can be instrumented directly within on-premises data centers or private clouds, capturing telemetry from custom AI models, application servers, and supporting infrastructure.

For API-based services, Dynatrace integrates seamlessly with cloud-native platforms, ingesting and correlating key performance indicators. This flexibility ensures that organizations benefit from unified observability regardless of deployment mode.



## Real-time insight into model behavior and performance

Dynatrace excels at detecting anomalies in critical AI model metrics, such as token usage, input/output length, latency, and cost per request. By continuously ingesting telemetry from every layer of the technology stack, Dynatrace applies advanced AI-powered analytics to monitor and correlate these signals in real time.

This enables teams to quickly identify deviations from normal patterns, such as unexpected spikes in token consumption, unusually long or short input/output sequences, latency fluctuations, or abnormal cost per request. Automated alerts and visualizations empower operators to respond proactively, minimizing downtime and optimizing resource allocation for generative AI applications.

Additionally, Dynatrace provides robust tracking of model versions, allowing organizations to maintain visibility into version changes and their impact on application performance. Teams can monitor the behavior of individual model variants and easily compare key metrics across different versions.

This capability supports efficient model A/B testing, enabling teams to evaluate the effectiveness of new models side by side. As a result, organizations can make informed decisions about model deployment.

## **Casual AI to surface root cause**

The Dynatrace Davis® AI engine automatically correlates performance and behavior anomalies with their underlying causes, enabling rapid and informed responses.

For instance, if there's a sudden spike in AI model hallucinations, Davis AI can trace the anomaly back to a recent version rollout or changes in input formation. This pinpointed context equips teams with actionable insights, allowing them to address issues quickly and confidently, reducing downtime and ensuring continued reliability of agentic applications.

## **Unified dashboards for observability and accountability**

Dynatrace offers a variety of pre-built views tailored to meet the needs of different users, including AI native developers, platform teams or site reliability engineers, and executives. These dashboards seamlessly combine infrastructure health metrics with AI model outputs and business impact indicators in a single, unified interface.

Developers can drill down into granular telemetry and debug model behavior, while SREs monitor system reliability and resource utilization alongside operational anomalies. Executives, meanwhile, gain high-level visibility into business KPIs and the direct effects of AI performance on customer experience, enabling data-driven decision making across the organization.

# Transform complexity into your greatest asset

The rise of generative AI marks a pivotal moment for every enterprise. Amazon Bedrock provides a powerful, flexible foundation for building and scaling AI-powered applications, but unlocking its full potential requires more than just adoption. It demands a new level of understanding – a unified view of your entire technology stack, from the underlying infrastructure to the cognitive behavior of AI models.

Without this clarity, you risk flying blind. Performance bottlenecks, security vulnerabilities, and rising costs can quickly undermine your AI initiatives, turning innovation into a source of frustration. But with the right observability strategy, you can turn this complexity into your greatest asset.

Dynatrace offers the end-to-end observability needed to master your AI-driven ecosystem. By providing unified insights into your application performance, model behavior, and business impact, Dynatrace empowers your teams to innovate with confidence. You can move from being reactive to proactive, automating intelligently and delivering digital experiences that are not only powerful but also reliable and secure.

**Ready to harness the full power of your AI investments?**

Contact us today for more information about how Dynatrace can help you maximize your Amazon Bedrock deployment.

#### ABOUT DYNATRACE

Dynatrace is advancing observability for today's digital businesses, helping to transform the complexity of modern digital ecosystems into powerful business assets. By leveraging AI-powered insights, Dynatrace enables organizations to analyze, automate, and innovate faster to drive their business forward. Learn more at [www.dynatrace.com](https://www.dynatrace.com).

