AWS re:Invent

NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SUP302

Troubleshooting from the perimeter

Mohammad Shafiullah (he/him)

Principal Cloud Support Engineer,
AWS Support Engineering
AWS

Venkata Bhavani Kanneganti (she/her)

Principal Cloud Support Engineer,
AWS Support Engineering
AWS

David Schliemann (he/him)

Principal Cloud Support Engineer,
AWS Support Engineering
AWS

Agenda

- ➢ Meet the AWS Support Engineering team
- ➢ Hear tips/methodologies for troubleshooting
- ➢ Q&A
- ➢ Participants will choose troubleshooting exercise track
- ➢ We will have some fun with the workshops!

AWS Support Engineering team

Mohammad

Bhavani

David

We know a thing or two about troubleshooting

Tips on troubleshooting at scale from the perimeter

When are we at the perimeter?

- ➢When our system has dependency on external system(s)
- ➢When the dependency is somewhat of a blackbox

AND (drumroll please)

- ➢When there are unknowns about our own system(s)

Tips on troubleshooting at scale from the perimeter

Why does troubleshooting from the perimeter need special consideration?

- ➢ You may not have visibility or access
- ➢ Troubleshooting becomes much easier with better observability
- ➢ You have to define your system's QoS and what is considered anomalous

Tips on troubleshooting at scale from the perimeter

You need the right setup/tooling to effectively troubleshoot at scale

Architectural awareness

- Client-side architecture
- Cloud infrastructure
- Data path
- Use cases

Configuration insights

- Configuration states trail
- Change management

Application instrumentation

- Appropriate tracers
- Clear metrics definition
- Delivery of tracing data

Low-level observability

- Appropriate profilers
- Clear metrics definition
- Delivery of tracing data

Canaries

- Canary distribution
- Baselines

Logs and triggers

- Unified search capability
- Retention policy
- Alarms

Troubleshooting methodologies primer

Plenty of methodologies exist in the industry and the ones we are mentioning are not new either!

1. Defining/understanding the problem
2. Use one or multiple methods (not an exhaustive list)
3. Perform a Correction of Errors to reduce the time to resolution in the future

Troubleshooting methodologies primer

Plenty of methodologies exist in the industry and the ones we are mentioning are not new either!

- 50/50 method
- Good vs. bad comparison method
- Hypothesis testing method
- Controlled reproductions method
- Building timelines method

Internet

One size does not fit all!

Troubleshooting methodologies primer

50/50 method

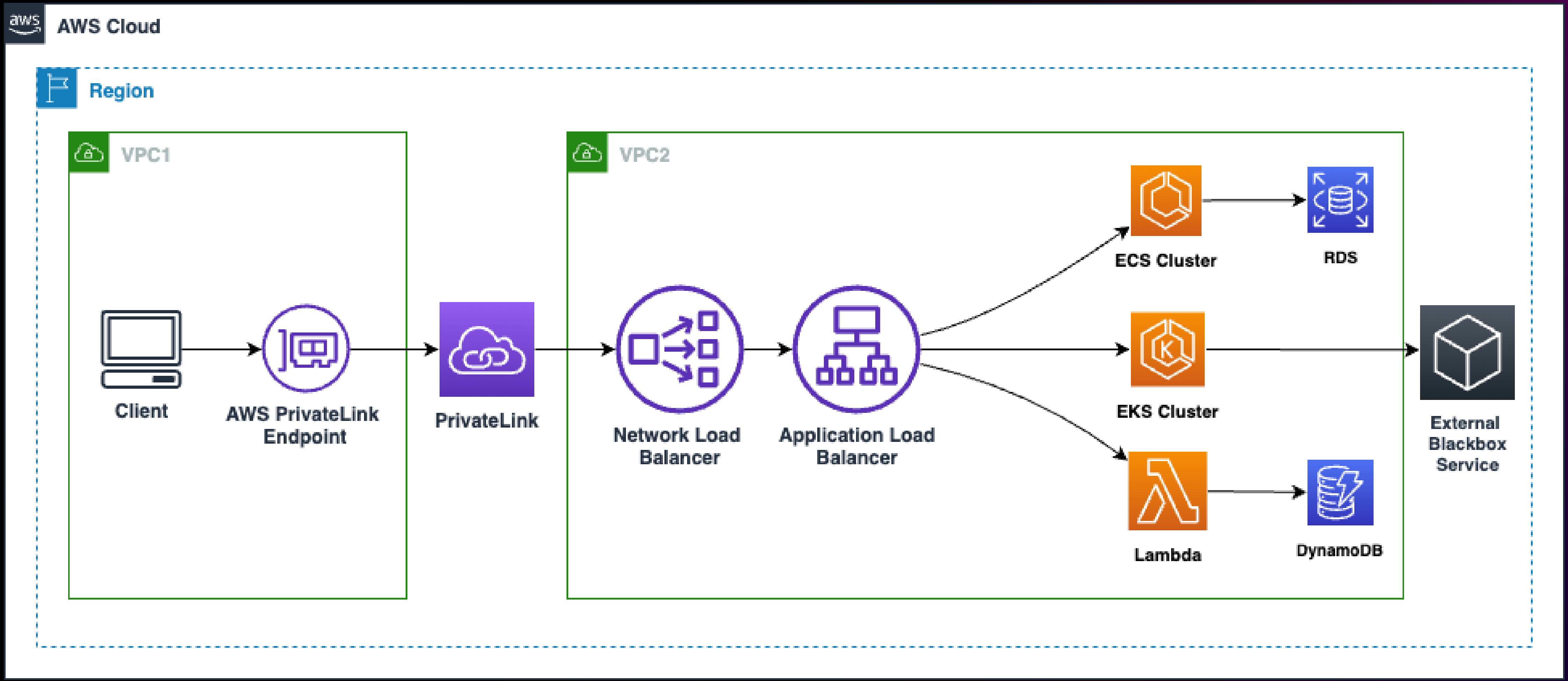
- More appropriate for cloud architectures without right instrumentation/monitoring

Internet

- Divide the end-to-end stack into two and cut down the pool of problematic sections by 50%

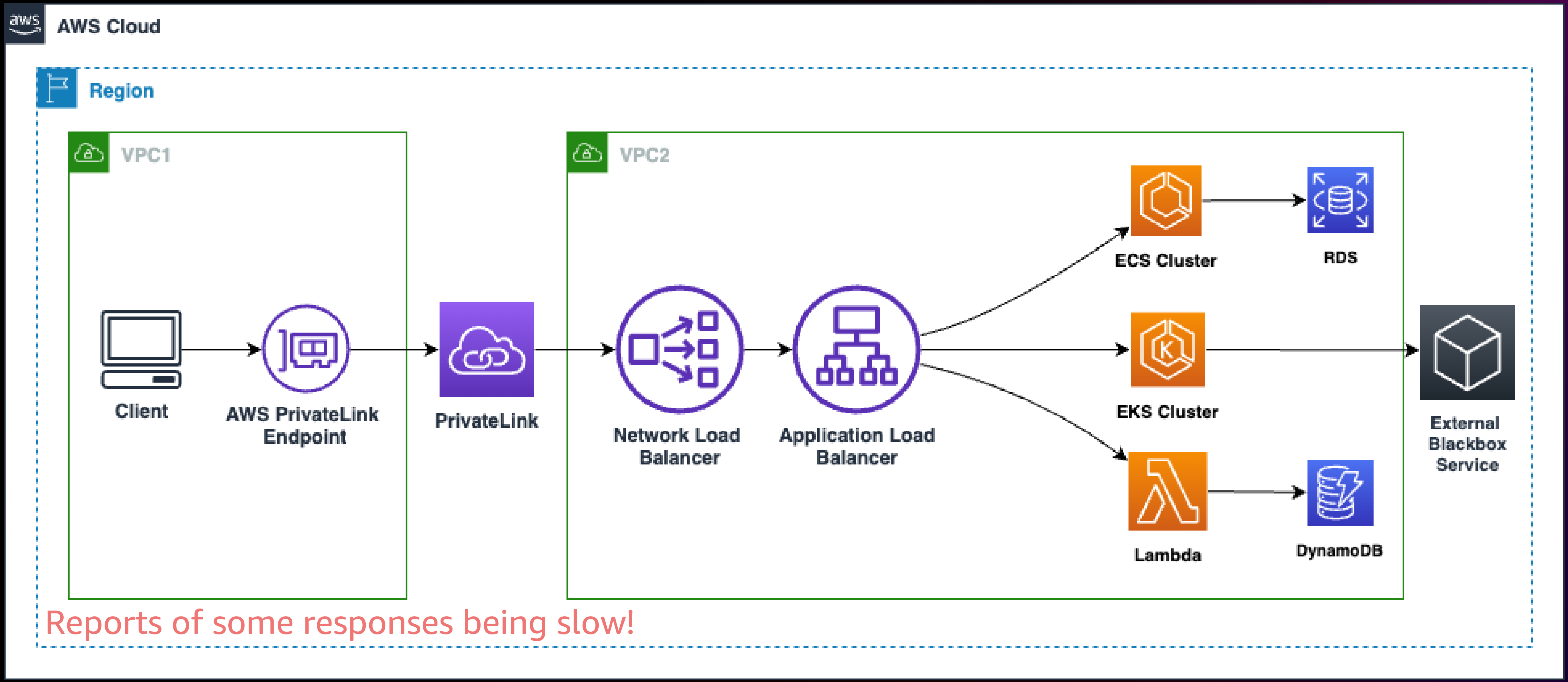
Troubleshooting methodologies primer

50/50 method



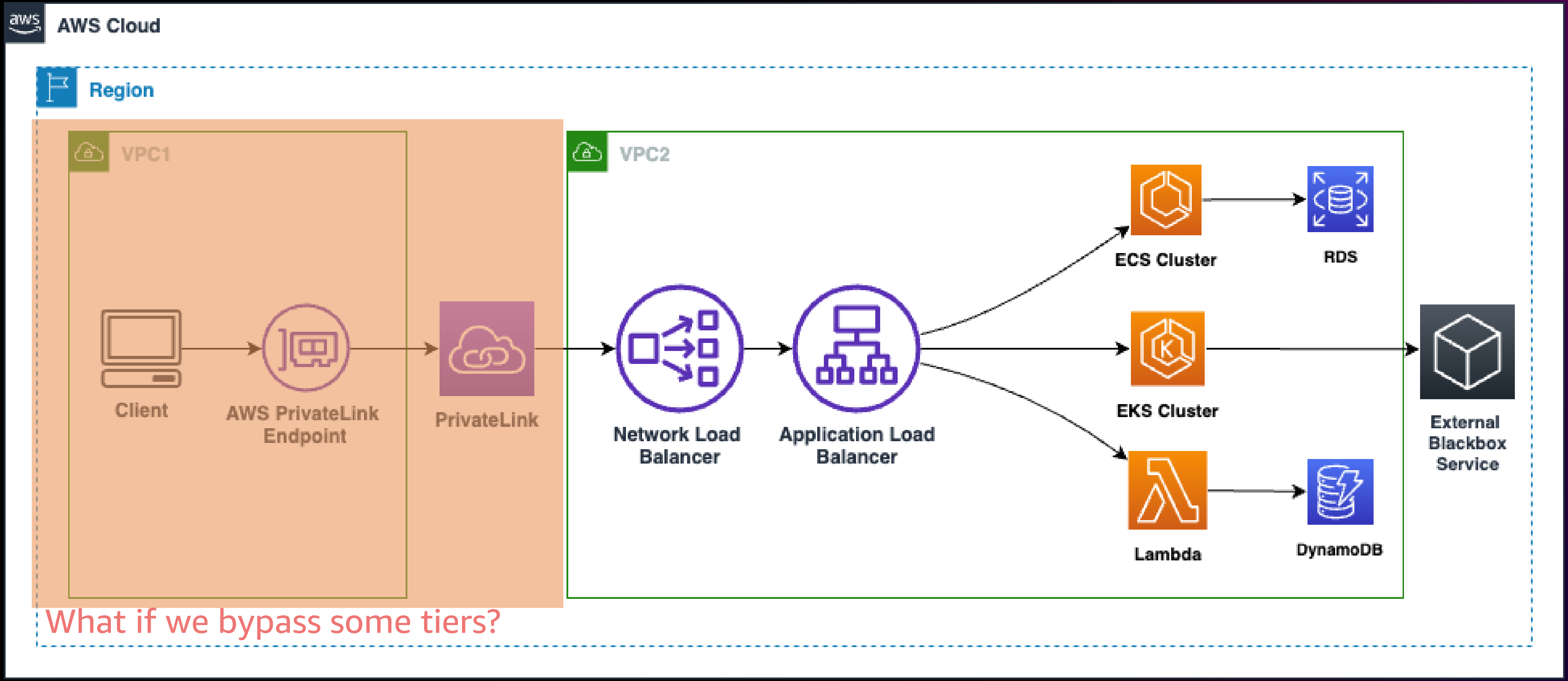
Troubleshooting methodologies primer

50/50 method



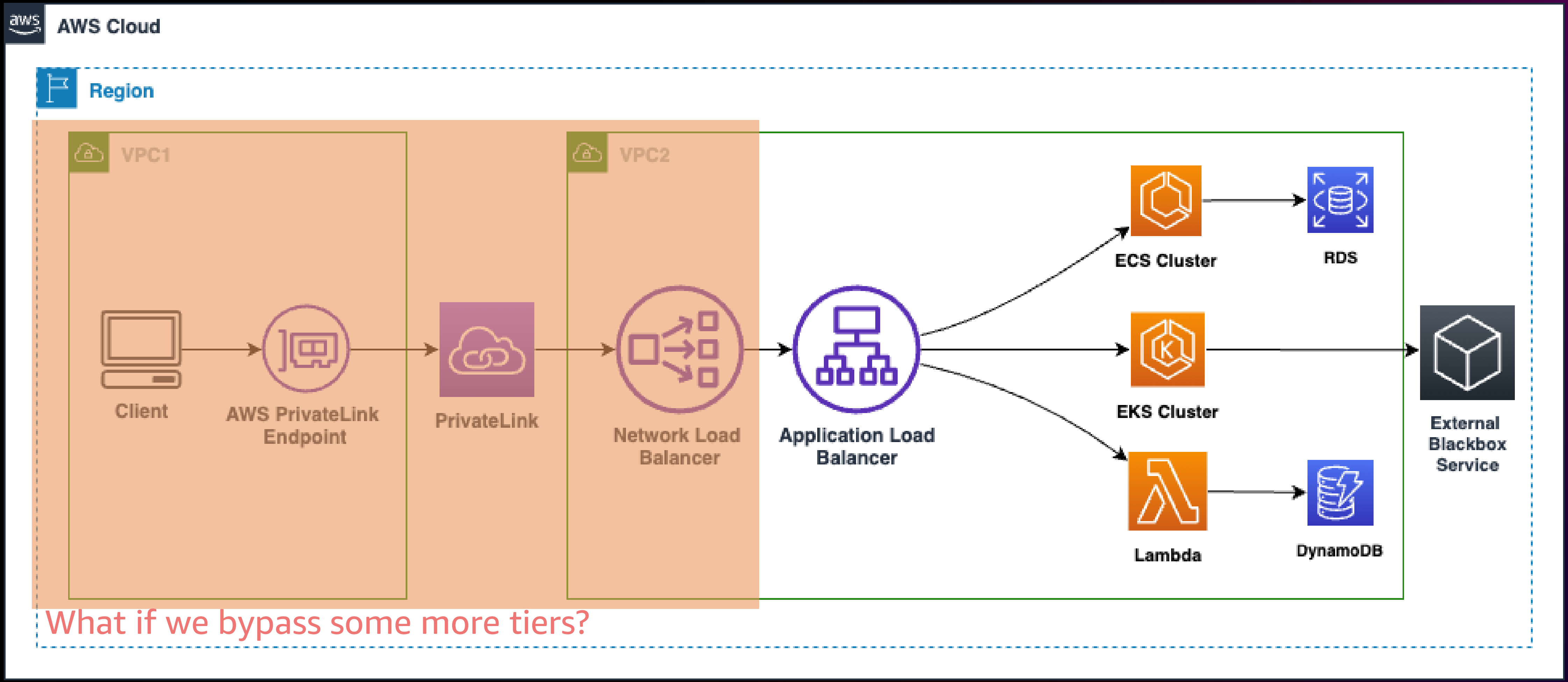
Troubleshooting methodologies primer

50/50 method



Troubleshooting methodologies primer

50/50 method



What if we bypass some more tiers?

Troubleshooting methodologies primer

Good vs. bad comparison method

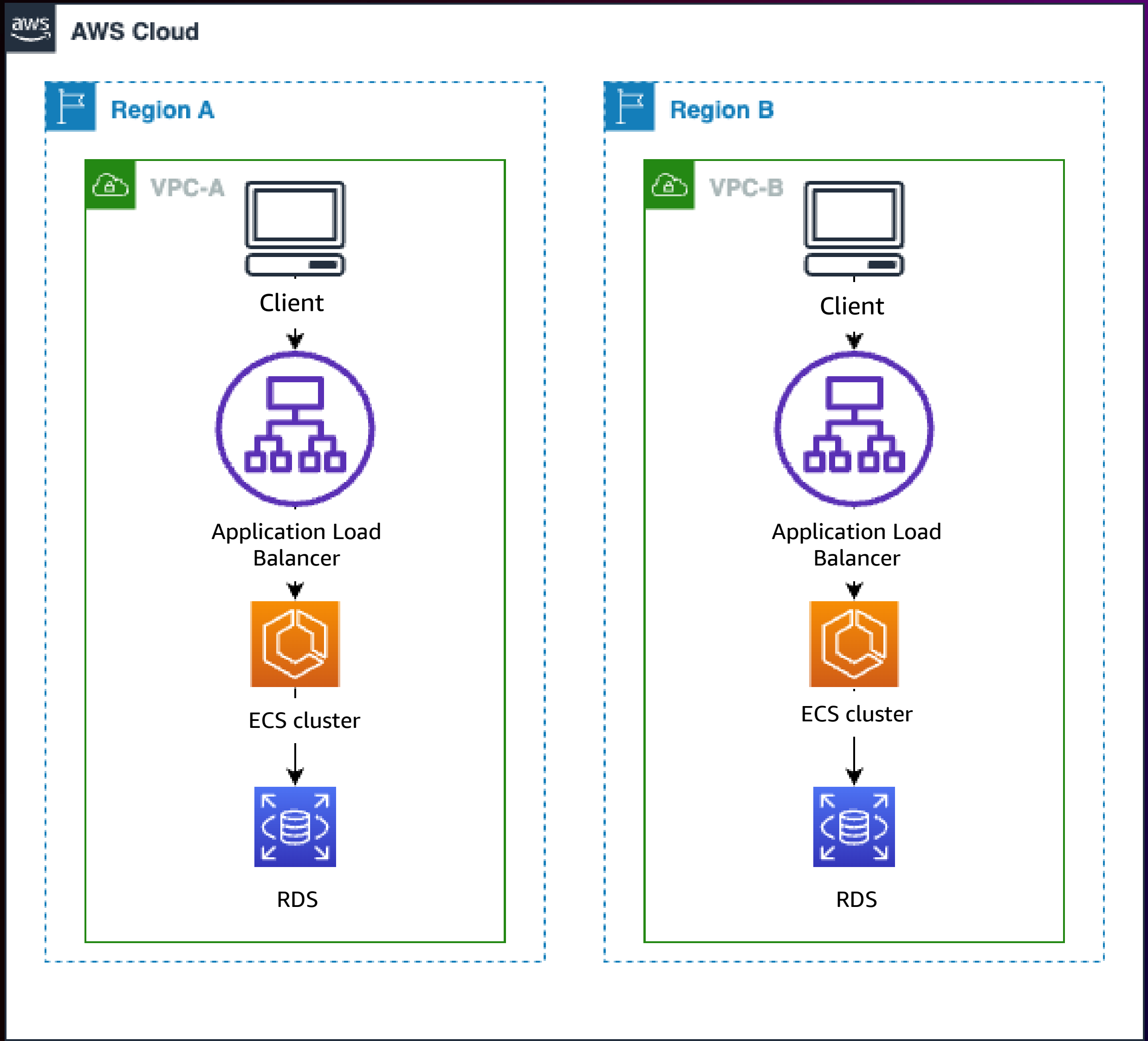
- Among multiple seemingly similar systems, one is having problems

Internet

- Compare the configuration and resource utilization between the working and not-working systems

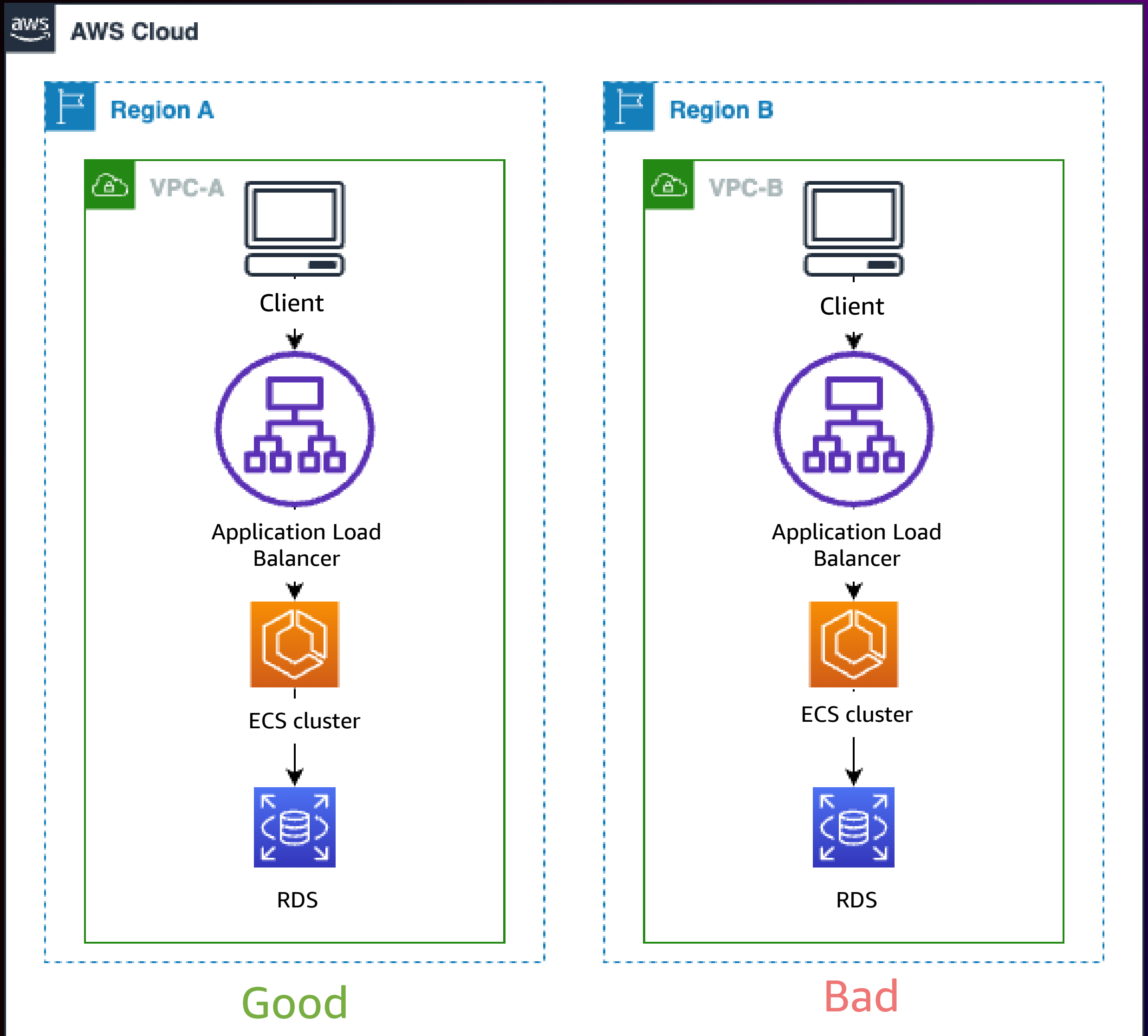
Troubleshooting methodologies primer

Good vs. bad comparison method



Troubleshooting methodologies primer

Good vs. bad comparison method

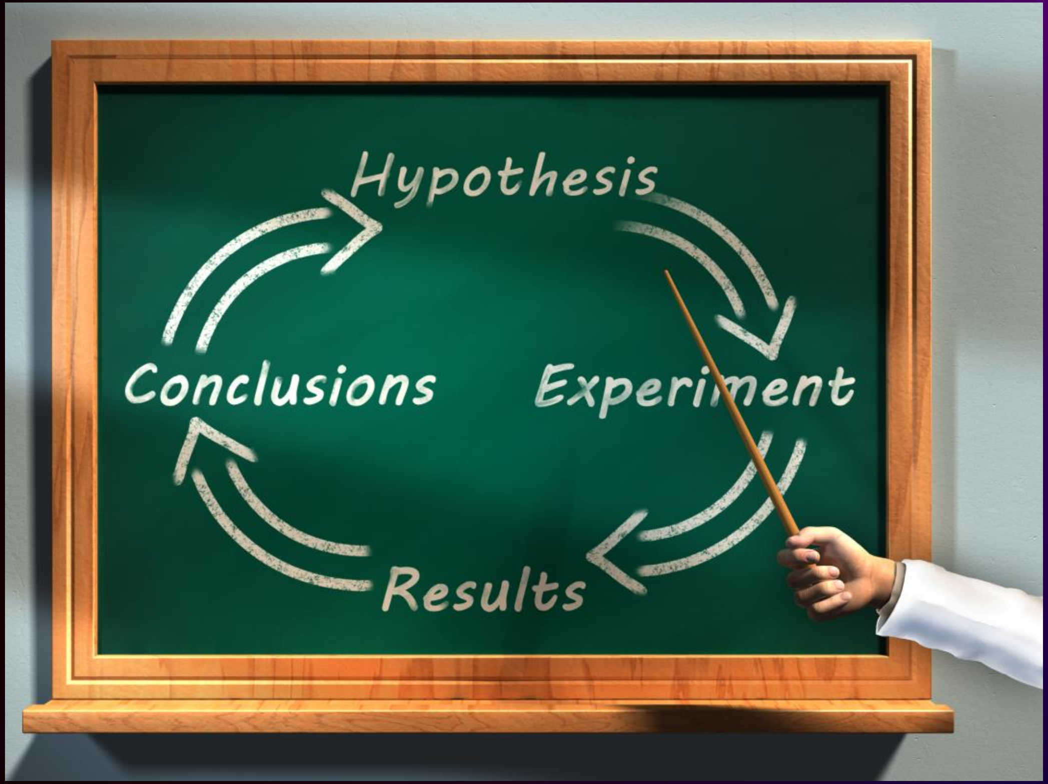


Troubleshooting methodologies primer

Hypothesis testing method

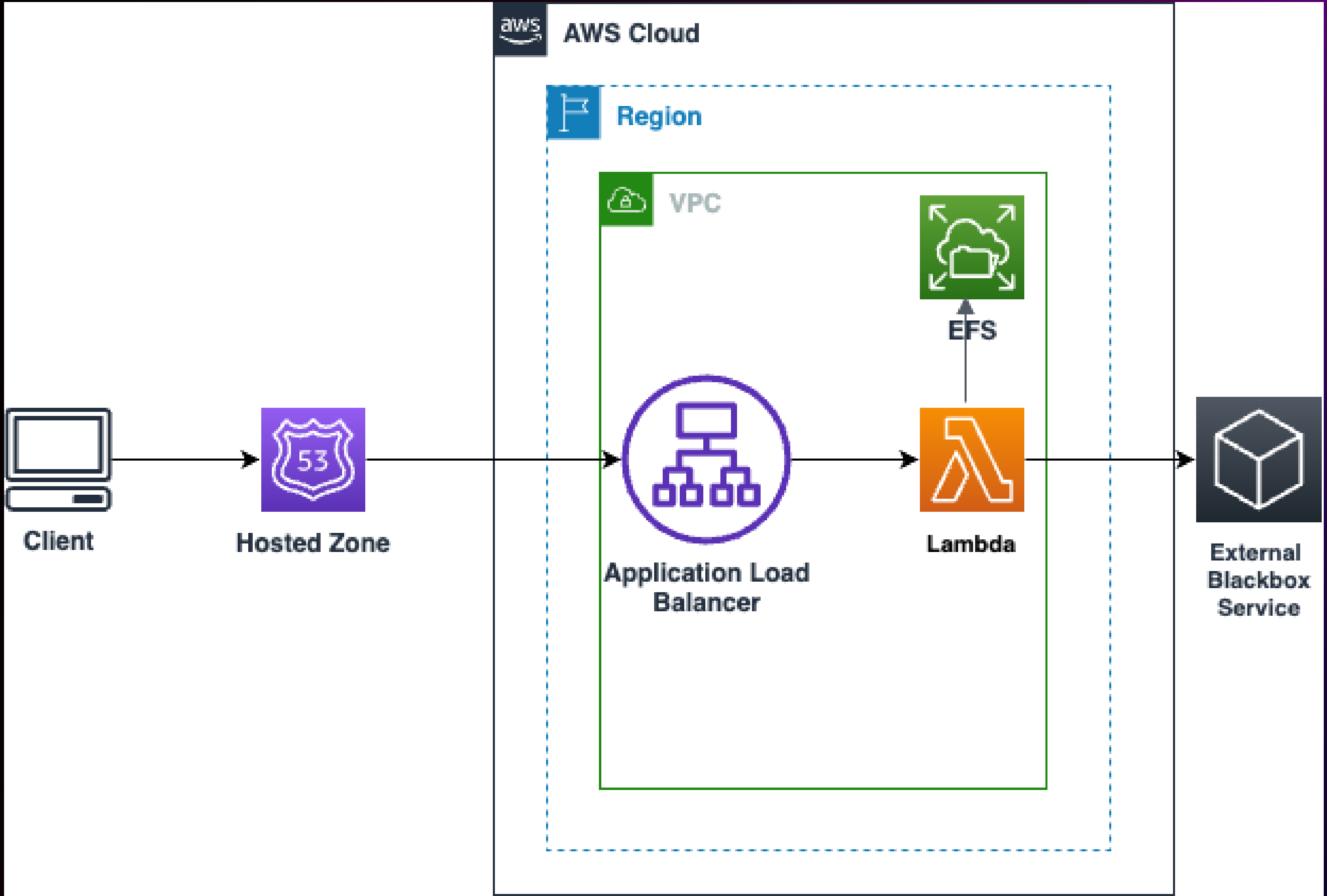
- Similar to the scientific method
- Come up with sets of hypotheses and investigate to prove or disprove each hypothesis
- Move on to next hypothesis and repeat

Internet



Troubleshooting methodologies primer

Hypothesis testing method

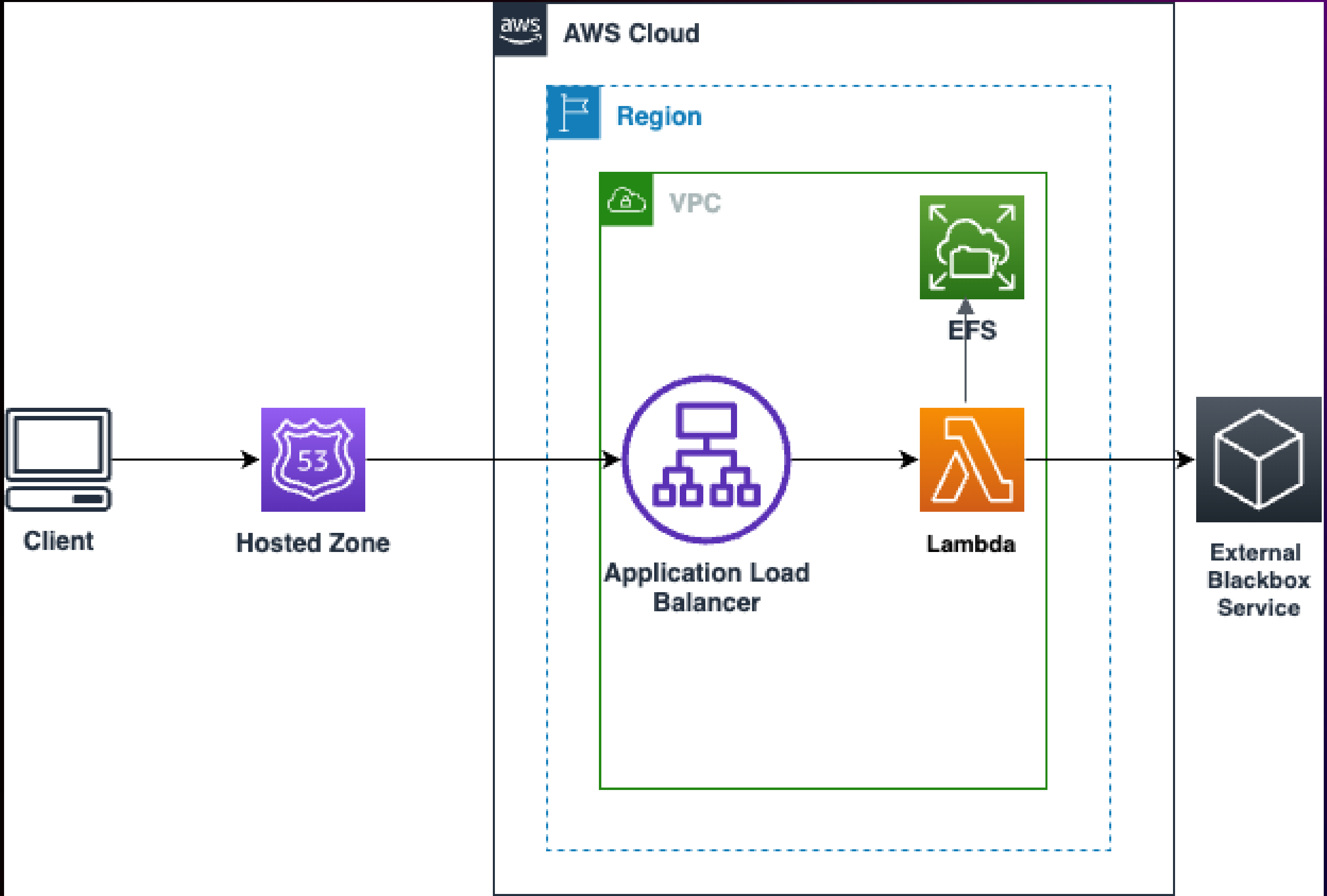


Troubleshooting methodologies primer

Hypothesis testing method

Slow response …

- DNS resolution taking longer
- External blackbox service is taking longer to respond
- …



Troubleshooting methodologies primer

Controlled reproduction method

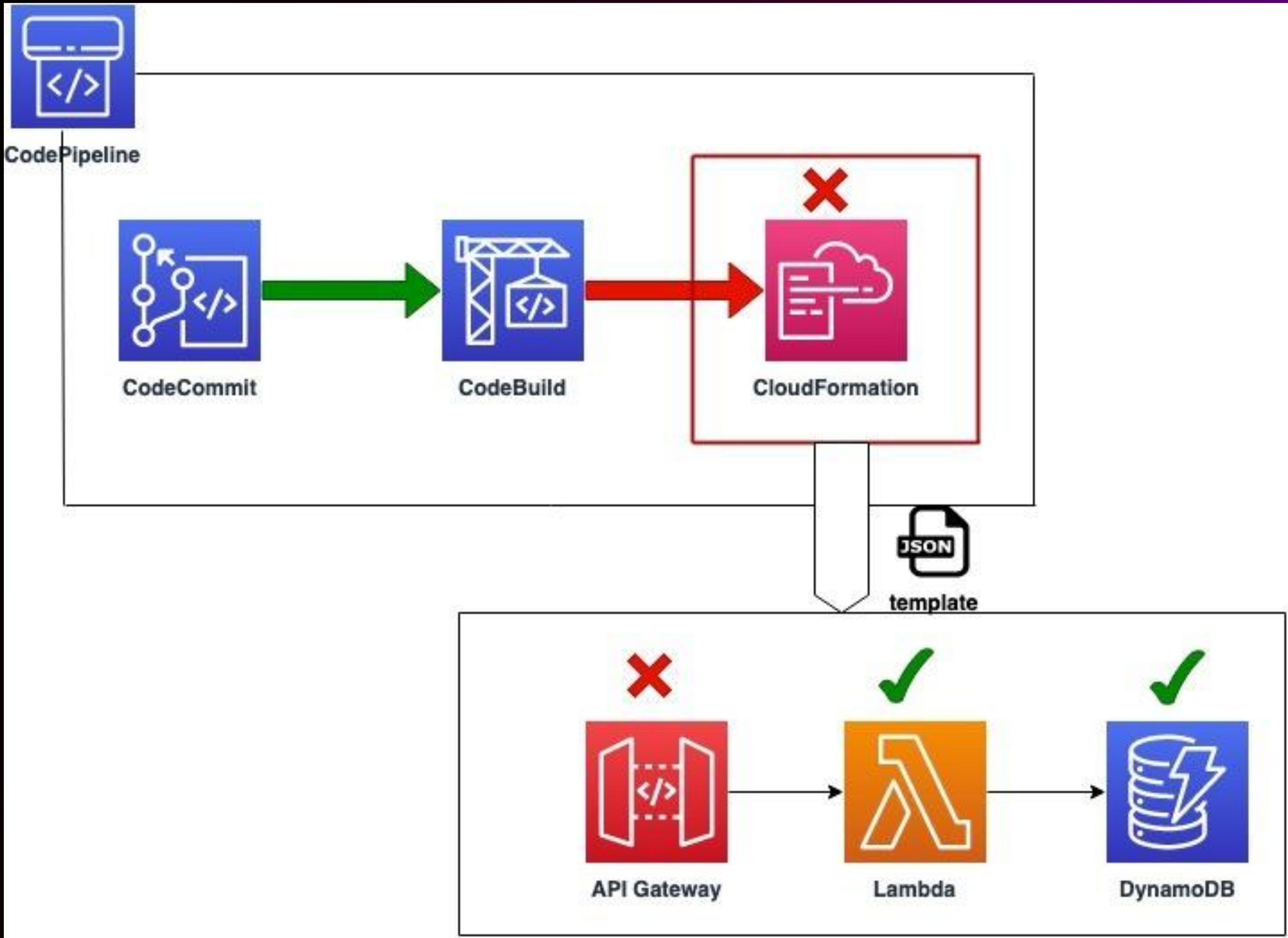
- Sometimes it is easier to attempt reproducing the issue in a controlled fashion
- You can set up an alternate resource for reproduction
- You can also simply use a sandbox environment to experiment

Troubleshooting methodologies primer

Controlled reproduction method

CodePipeline failed deployment…

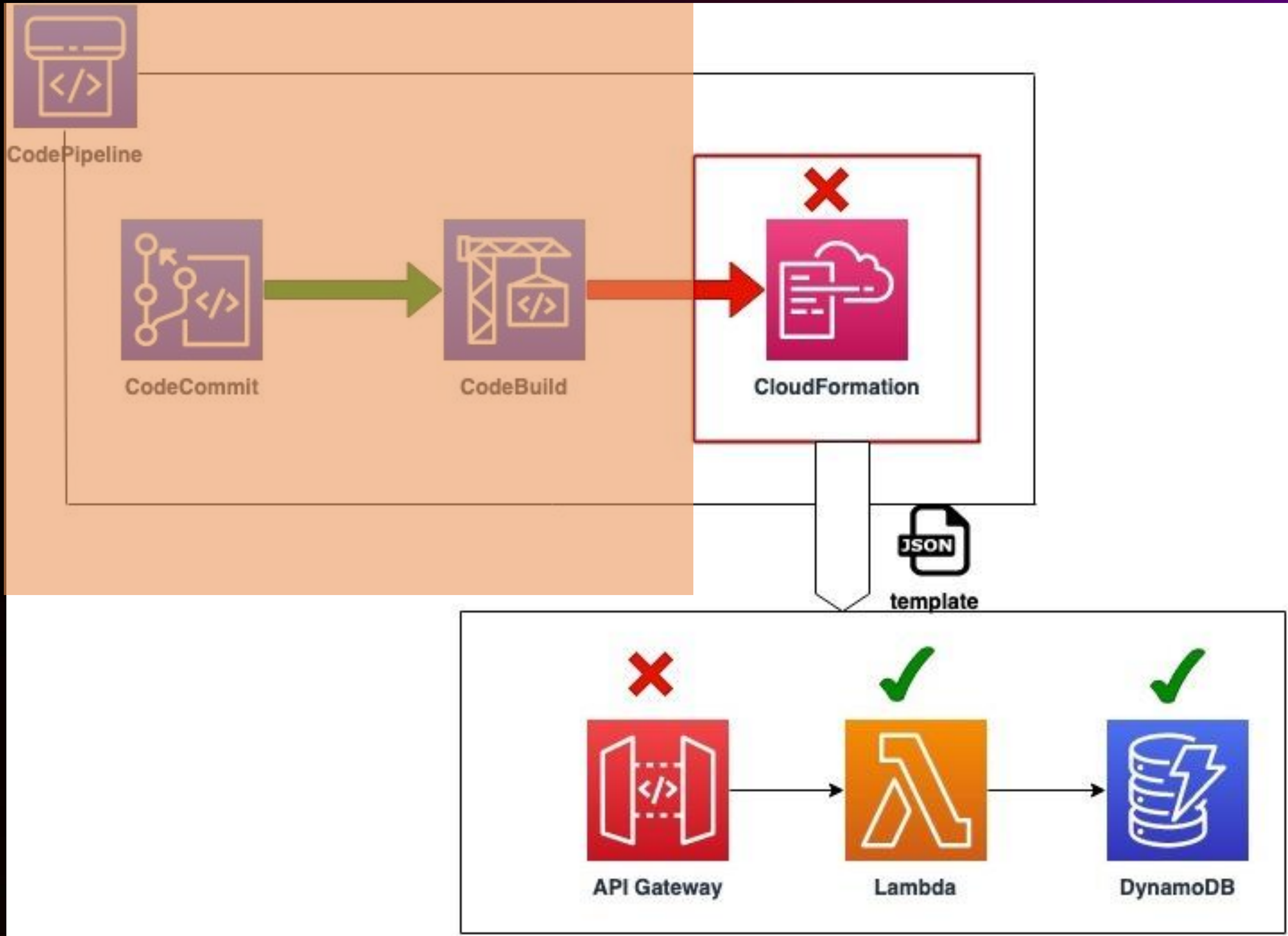
- CFN failed to create serverless infrastructure
- In particular, provisioning API Gateway failed



Troubleshooting methodologies primer

Controlled reproduction method

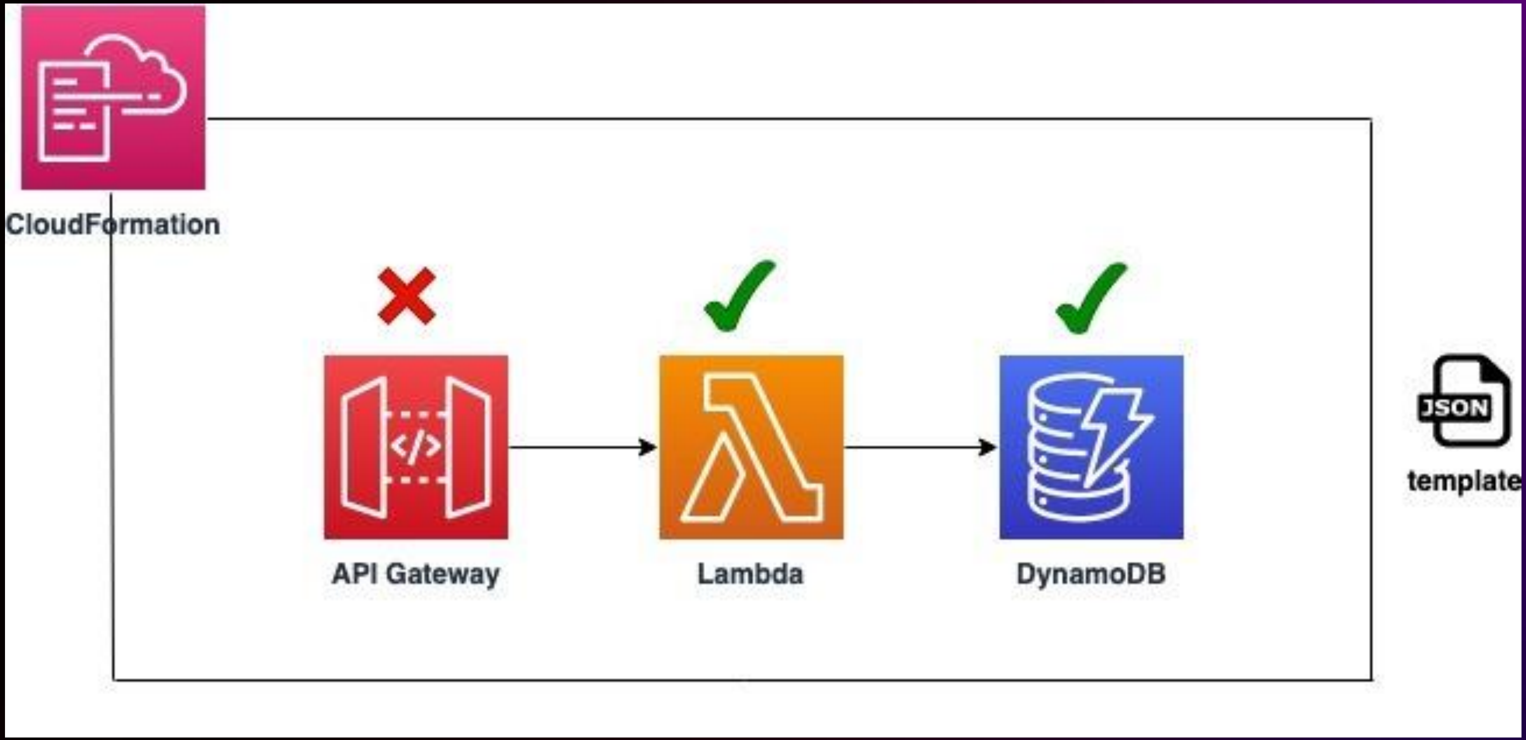
We could reproduce the failure in a different account with just CloudFormation attempting to create stack



Troubleshooting methodologies primer

Controlled reproduction method

We could reproduce the failure in a different account with just CloudFormation attempting to create stack



Troubleshooting methodologies primer

Building timelines method

- “When did the issue start?” and “What changed?” are valuable questions if they can be answered precisely and objectively
- Overlay configuration changes with issue start/duration
- Often used to rollback changes to previous working state and then investigate what happened

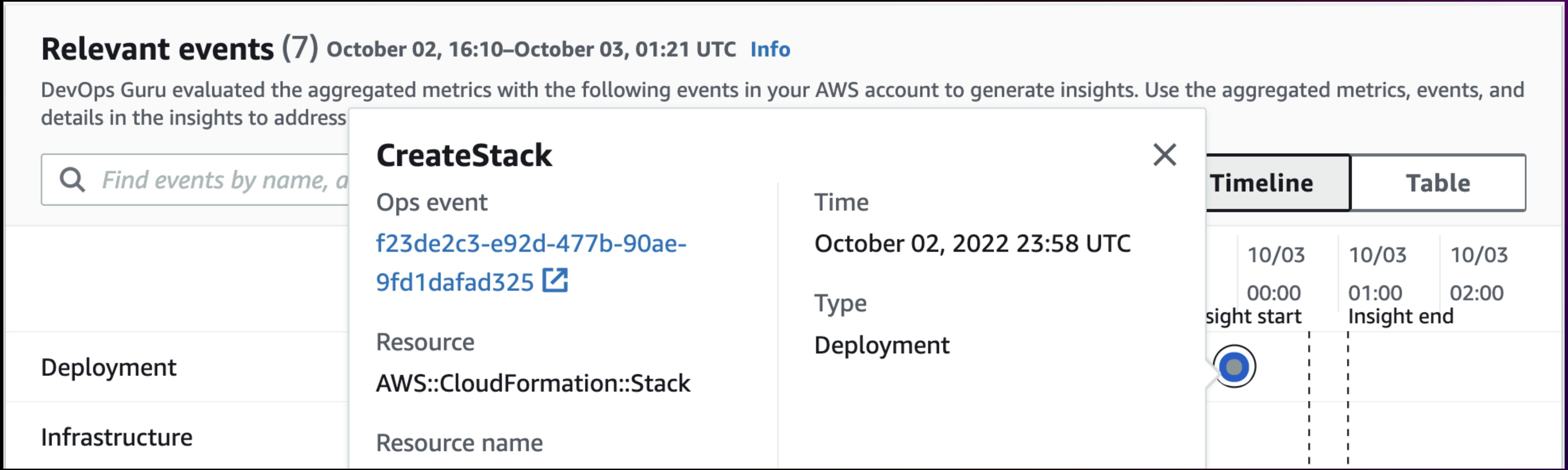
Amazon DevOps Guru

AWS X-Ray

Troubleshooting methodologies primer

Building timelines method

Amazon DevOps Guru



Relevant events (7) October 02, 16:10–October 03, 01:21 UTC Info

DevOps Guru evaluated the aggregated metrics with the following events in your AWS account to generate insights. Use the aggregated metrics, events, and details in the insights to address

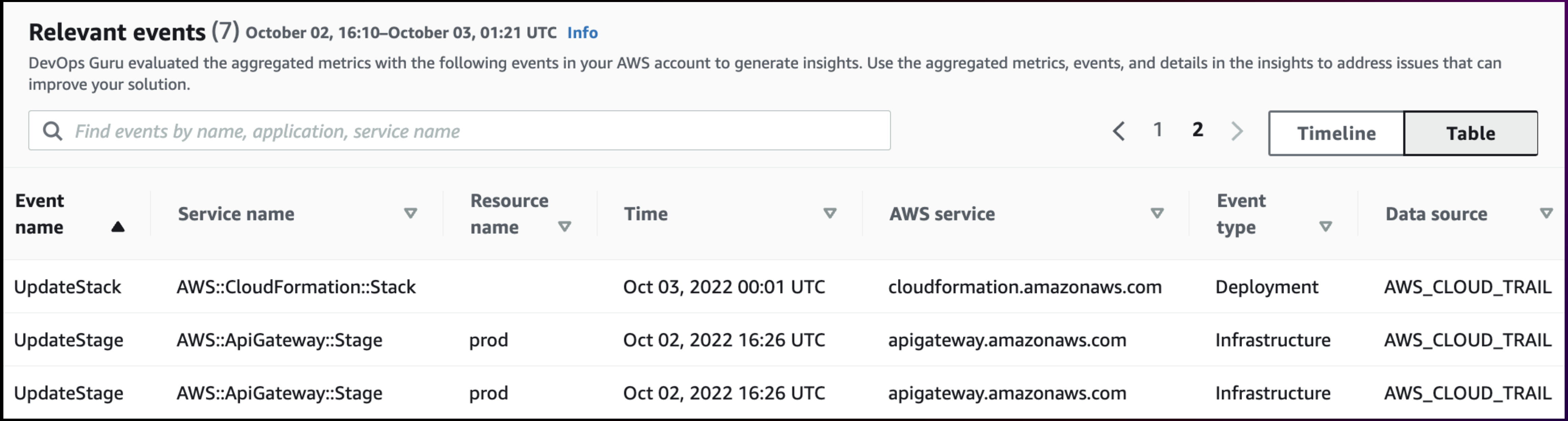
CreateStack

Ops event	Time
f23de2c3-e92d-477b-90ae-9fd1dafad325	October 02, 2022 23:58 UTC
Resource	Type
AWS::CloudFormation::Stack	Deployment
Resource name	

Troubleshooting methodologies primer

Building timelines method

Amazon DevOps Guru



Relevant events (7) October 02, 16:10–October 03, 01:21 UTC Info

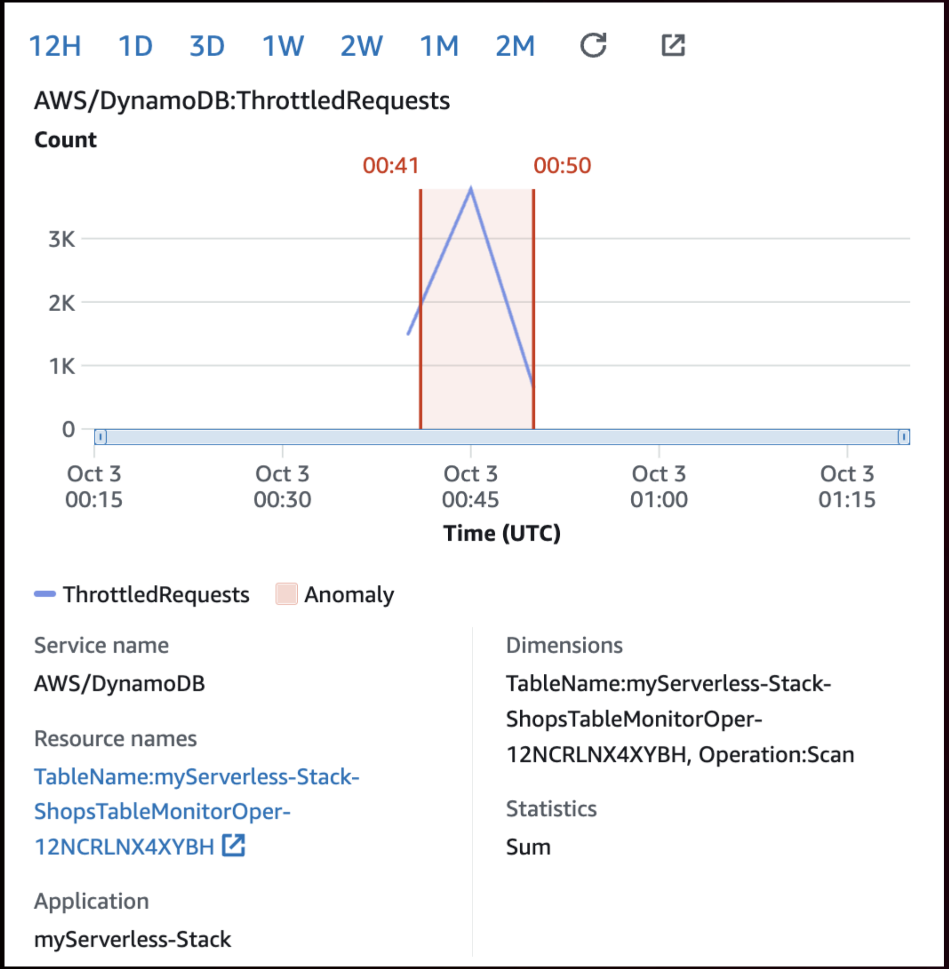
DevOps Guru evaluated the aggregated metrics with the following events in your AWS account to generate insights. Use the aggregated metrics, events, and details in the insights to address issues that can improve your solution.

Event name	Service name	Resource name	Time	AWS service	Event type	Data source
UpdateStack	AWS::CloudFormation::Stack		Oct 03, 2022 00:01 UTC	cloudformation.amazonaws.com	Deployment	AWS_CLOUD_TRAIL
UpdateStage	AWS::ApiGateway::Stage	prod	Oct 02, 2022 16:26 UTC	apigateway.amazonaws.com	Infrastructure	AWS_CLOUD_TRAIL
UpdateStage	AWS::ApiGateway::Stage	prod	Oct 02, 2022 16:26 UTC	apigateway.amazonaws.com	Infrastructure	AWS_CLOUD_TRAIL

Troubleshooting methodologies primer

Building timelines method

Amazon DevOps Guru



AWS/DynamoDB:ThrottledRequests

Service name
AWS/DynamoDB

Resource names
TableName:myServerless-Stack-ShopsTableMonitorOper-12NCRLNX4XYBH

Application
myServerless-Stack

Dimensions
TableName:myServerless-Stack-ShopsTableMonitorOper-12NCRLNX4XYBH, Operation:Scan

Statistics
Sum

Troubleshooting methodologies primer

You should:

- ➢ Pay close attention to numbers that are indicative of a pattern
- ➢ Exercise the 5 Whys once the dust settles

Internet

- ➢ Understand your P’s (P50, P90, P99, P99.9, etc.)

Troubleshooting methodologies primer

You should:

- ➢ Ask for transparency from internal and external teams (improved logging, granular metrics, SOPs, etc.)
- ➢ Ask for transparency on how these metrics are measured

Internet

- ➢ Trust but verify

Troubleshooting methodologies primer

You should avoid:

- Confirmation bias – not challenging assumptions
- Random trial and error without keeping track of what has been tested so far

Internet

- Changing multiple things at a time, randomly hoping for the best
- Drawing a correlation between two symptoms or events that coincided when the correlation is not backed by data

Troubleshooting methodologies primer

You should avoid:

- Reversing cause and effect
- Looking only where you have metrics – the problem can reside elsewhere

Internet

- Delaying the creation of an observability infrastructure until there is an issue

Reminder: Tips on troubleshooting at scale from the perimeter

You need the right setup/tooling to effectively troubleshoot at scale

Architectural awareness

- Client-side architecture
- Cloud infrastructure
- Data path
- Use cases

Configuration insights

- Configuration states trail
- Change management

Application instrumentation

- Appropriate tracers
- Clear metrics definition
- Delivery of tracing data

Low-level observability

- Appropriate profilers
- Clear metrics definition
- Delivery of tracing data

Canaries

- Canary distribution
- Baselines

Logs and triggers

- unified search capability
- Retention policy
- Alarms

Troubleshooting Labs

- ➢ Three separate lab streams you can choose from:
  - Stream 1: Networking and web services
  - Stream 2: DevOps and serverless
  - Stream 3: Database
  - ➢ Obstacle-course-like labs
    We will incorporate some of our troubleshooting suggestions from the presentation

AWS re:Post

Community-driven knowledge sharing site for your AWS questions
Visit: https://repost.aws/

Login is not required to browse content

Content is organized by topics and tags

Machine learning drives smart content recommendations

Follow specific questions, topics, and community experts

Additional resources

[AWS re:Post]

Community-driven knowledge sharing site

Thank you!

Please complete the session survey in the **mobile app**