

edureka!



Microsoft Azure DevOps Solution Certification (AZ-400)

COURSE OUTLINE



Azure AZ-400

MODULE 1: Introduction to Azure DevOps

MODULE 2: Implementing Continuous Integration

MODULE 3: Build Containers with Azure DevOps

MODULE 4: Designing a Dependency Management Strategy and Managing Artifact Versioning

Artifact Versioning

MODULE 5: Setting up Release Management Workflow

MODULE 6: Implementing Deployment Models and Services

MODULE 7: Implement and Optimize Continuous Feedback Mechanism

MODULE 8: Azure Tools: Infrastructure and Configuration, and Third-Party Tools

MODULE 9: Implementing Compliance and Security

MODULE 10: Azure Case Studies

Implement and Optimize Continuous Feedback Mechanisms

Topics

Following are the topics covered in this module:

- Tools to Track System Usage, Feature Usage, and Flow
- Azure Dashboard
- Ticketing Systems
- Practices for Measuring End-User Satisfaction
- Processes for Capturing and Analyzing user Feedback
- Process for Automating Application Analytics
- Site Reliability Engineering
- Baseline by Analyzing Telemetry
- Tuning to Reduce Meaningless or Non-Actionable Alerts

Objectives

After completing this module, you should be able to:

- Configure and create Azure DevOps dashboard
- Integrate and configure ticketing systems
- Design practices for measuring end-user satisfaction
- Capture and analyze user feedback
- Establish a baseline by analyzing telemetry
- Demonstrate how to monitor application performance
- Perform integration between Azure DevOps and Teams





Collaborating in Pandemic

Microsoft and Global Pandemic



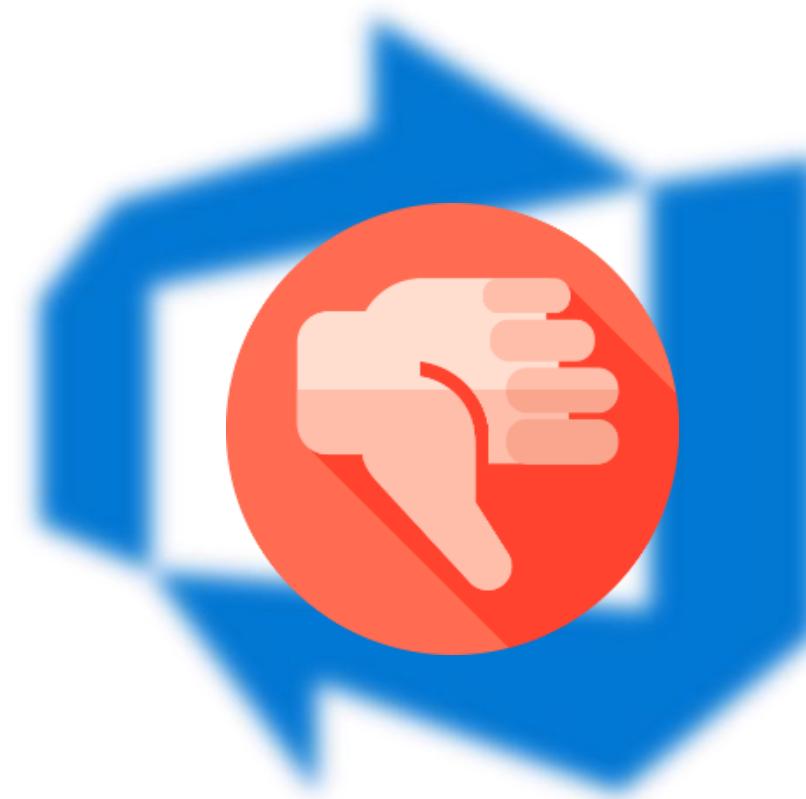
Due to the pandemic, Microsoft has asked all its employees to work from home

Monitoring the Application

Microsoft usually monitors the applications performance and output parameters through Azure DevOps

THE ISSUE

But it becomes difficult to monitor the application's performance through Azure DevOps when all the employees are working remotely. So, Microsoft hired Mike as a DevOps Administrator to address this



Azure DevOps

The Next Action

Mike's initial job is to work collaboratively with the team and find out a way to monitor the application's performance in real-time, and then visually monitor the output parameters



Mike
DevOps Administrator

The Research

We can use Microsoft Teams for this task. That way, all the users can get the DevOps pipeline status



Now you know the challenge.
What would you suggest?

The Solution



- We have to integrate Azure DevOps with Team and monitor the application's performance in real-time
- Also, set up a visualization technique in Teams to visually monitor the output parameters

Azure DevOps with Teams



- Azure DevOps Services integration with Microsoft Teams offers a comprehensive chat and collaborative interface throughout the development cycle.
- Teams can easily stay aware of essential activities in your Azure DevOps team projects with updates and alerts on work objects, pull requests, code commits, create and release



Implementing Tools to Track System Usage, Feature Usage, and Flow

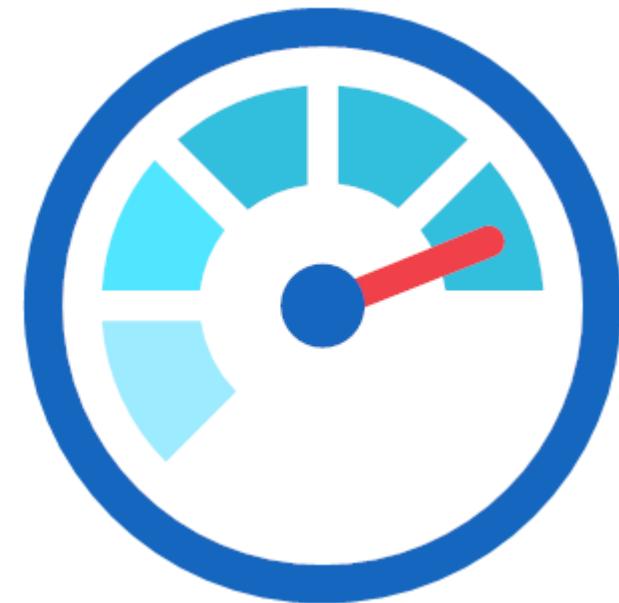
Continuous Monitoring



Continuous monitoring refers to processes and technologies used to monitor each phase of the application's lifecycle. Azure Monitor and Application Insights are used for this

Azure Monitor: Overview

- Azure Monitor collects the data from live applications
- It helps in tracking the health and performance of the entire infrastructure, including virtual machines, containers, storage, and networks



What does Azure Monitor Collect?

Azure monitor helps collect:

01

Platform metrics, activity logs, and diagnostics logs

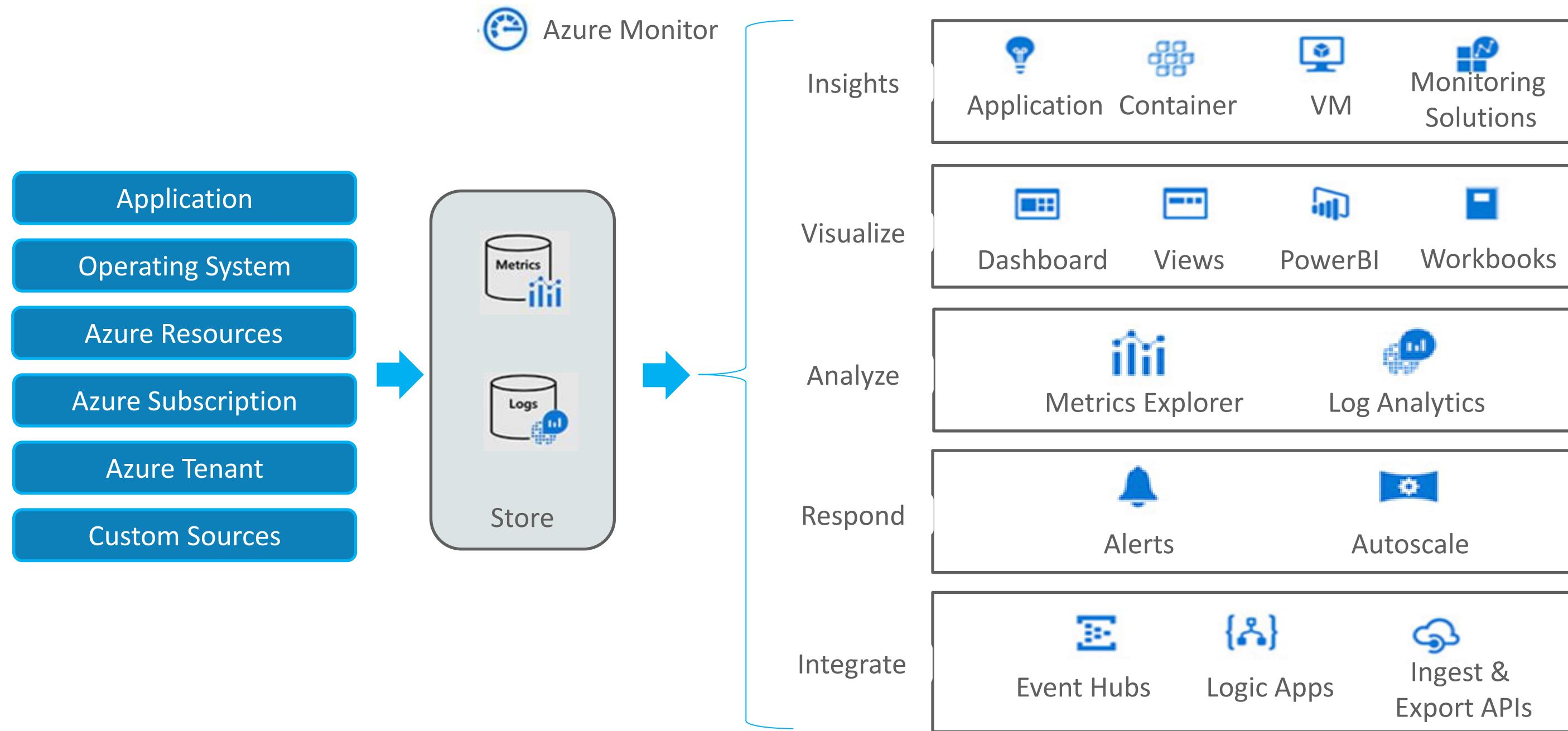
02

Monitoring Data from virtual machines

03

Monitoring data for containers or Kubernetes cluster

High-level View of Azure Monitor



Alerts through Azure Monitor

Alerts can be defined based on the data collected. Some examples of metrics where alerts can be defined are:

- 1 Metric values
- 2 Log search queries
- 3 Activity log events
- 4 The health of the underlying Azure platform
- 5 Tests for website availability

Azure Application Insights: Overview

Azure Application Insights is an azure service used for:

Monitoring web application

Automatically detecting performance anomalies

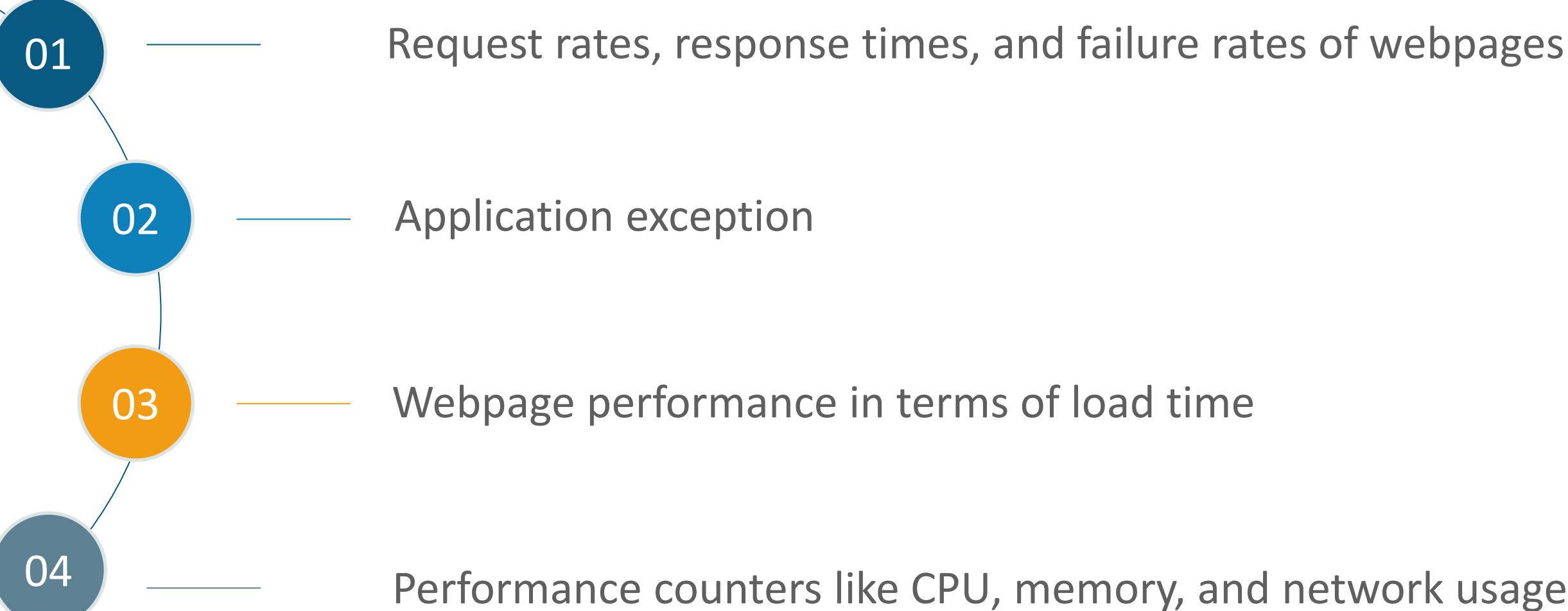
Diagnosing failure

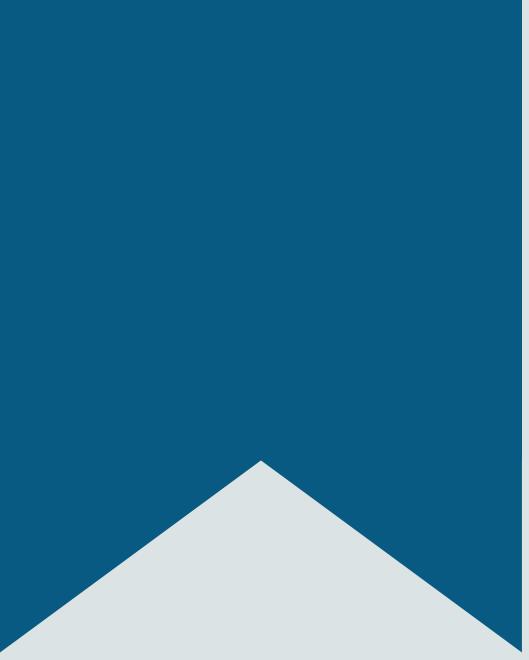
Understanding the user behaviors through their actions



What does Azure Application Insights Monitor?

Azure Application insights can monitor:

- 
- 01 Request rates, response times, and failure rates of webpages
 - 02 Application exception
 - 03 Webpage performance in terms of load time
 - 04 Performance counters like CPU, memory, and network usage

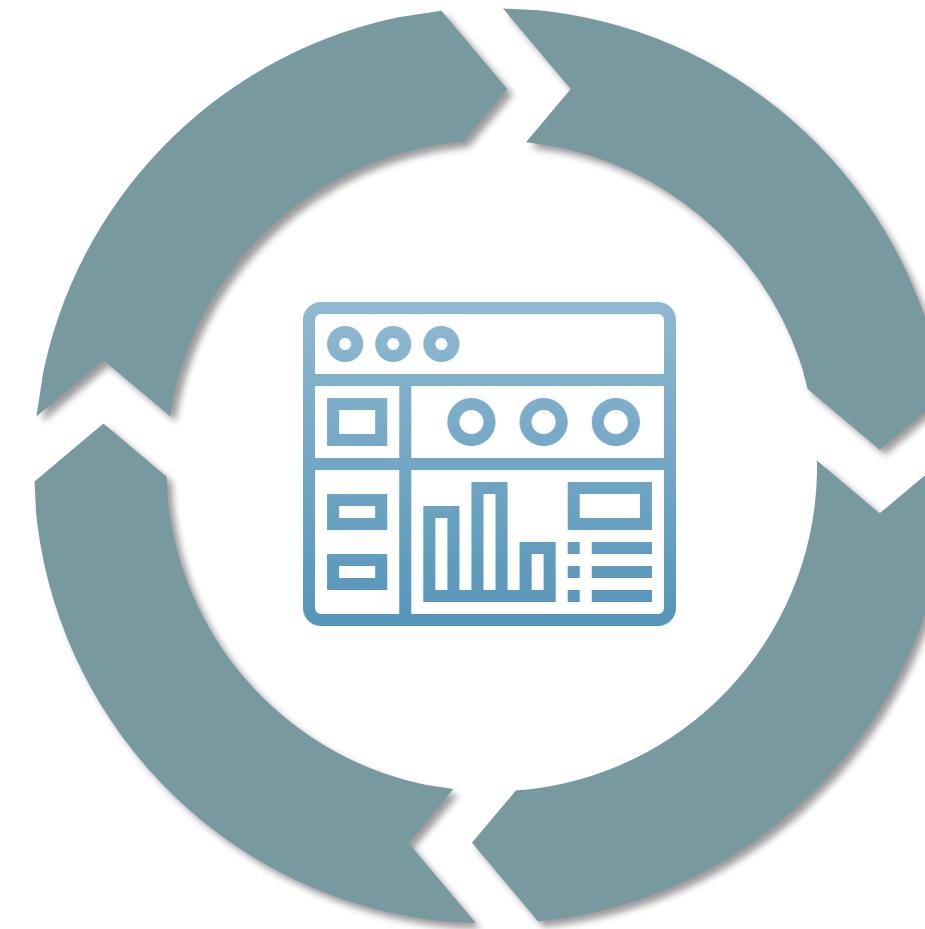


Azure Dashboard

Azure DevOps Dashboard

The Azure DevOps Dashboard is a highly configurable dashboard that allows you to:

Enhance workflow process
and flexibility



Share Information

Monitor Progress

Monitor Trends

Features of Azure Dashboard

It is a customizable interactive signboard that provides real-time information

Widget is a catalog that displays the information

These reports are derived from available information

Dashboard

Charts

Widgets

Power BI Reports

In-Context Report

Results can be shown in charts for better visualization

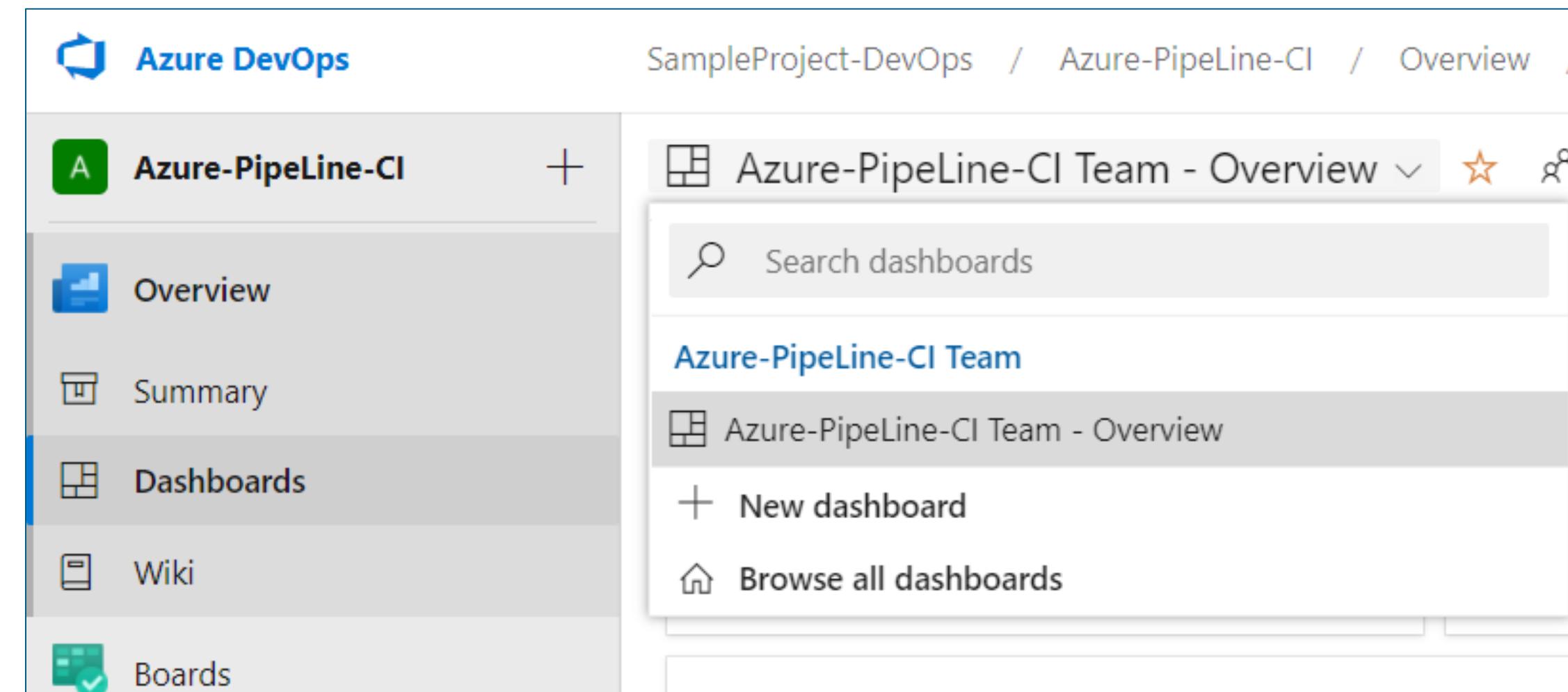
Users can create a Power BI report using Analytics data

Dashboard Creation: Step 1

Dashboard creation consists of the following steps:

Open Azure DevOps and navigate to any working project

Go to Overview → Dashboards and click on the dropdown as shown here:



Dashboard Creation: Step 2

Click on New Dashboard and fill in the information as shown below. The dashboard can be created for team or project

Create a dashboard

Name*
ProjectDashboard

Description
Project Dashboard

Automatically refresh the dashboard every 5 minutes

Dashboard Type

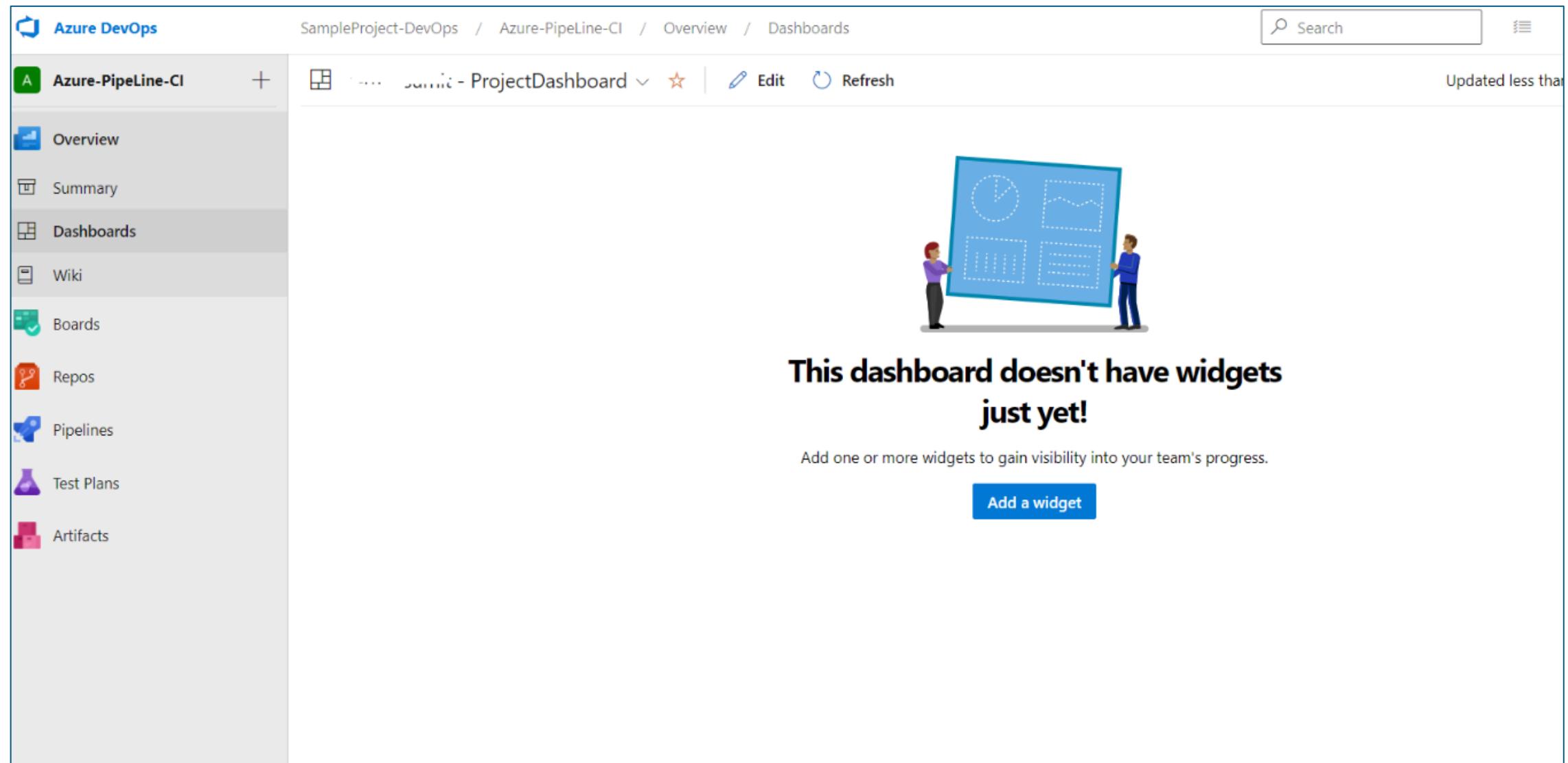
Team Dashboard
The dashboard is associated with a single team. Team admins can edit and manage this dashboard. Everyone can view the dashboard.

Project Dashboard
The dashboard is not associated with a team. You decide which users and groups can edit and manage this dashboard. Everyone can view the dashboard.

Create **Cancel**

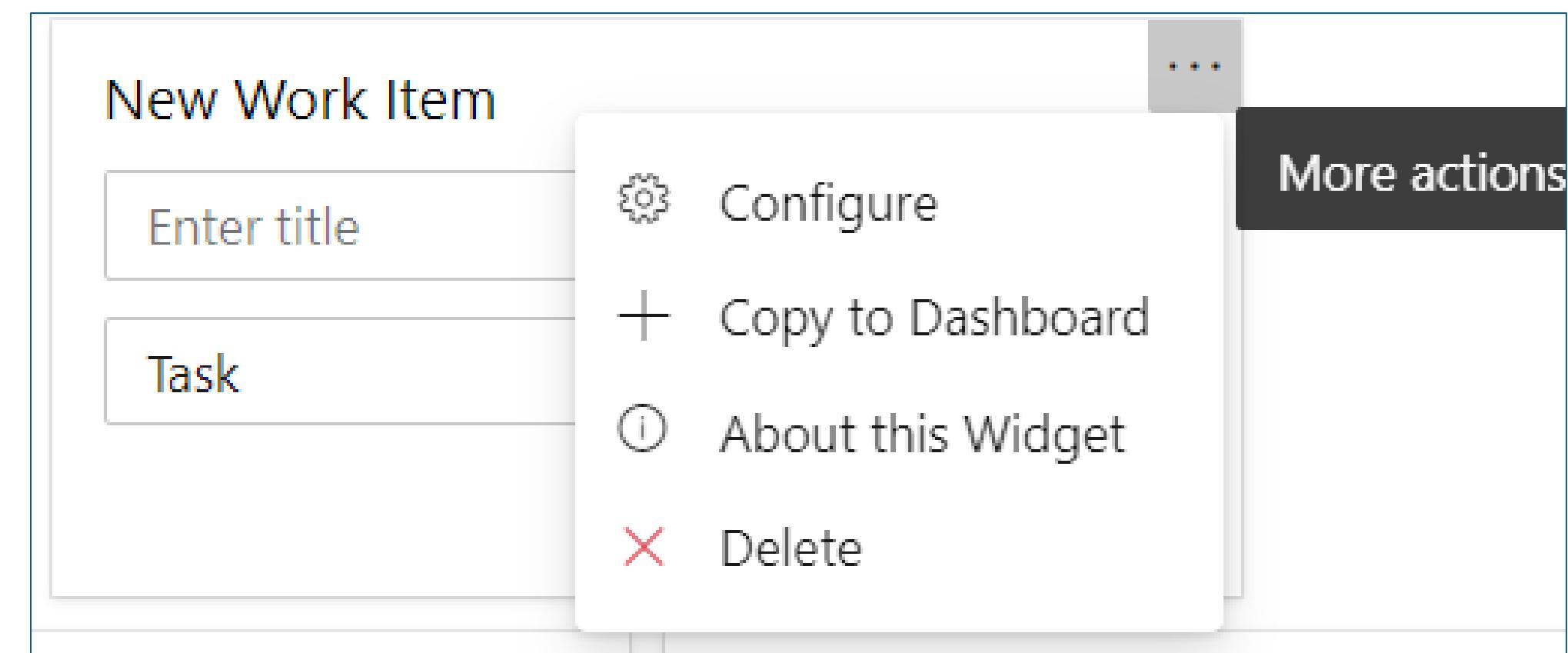
Dashboard Creation: Step 3

The following screen will pop-up. Click on Create Widget



Dashboard Creation: Step 4

Select the widget and configure the connection. It can be done by clicking on the "..." button on the top right side of each widget

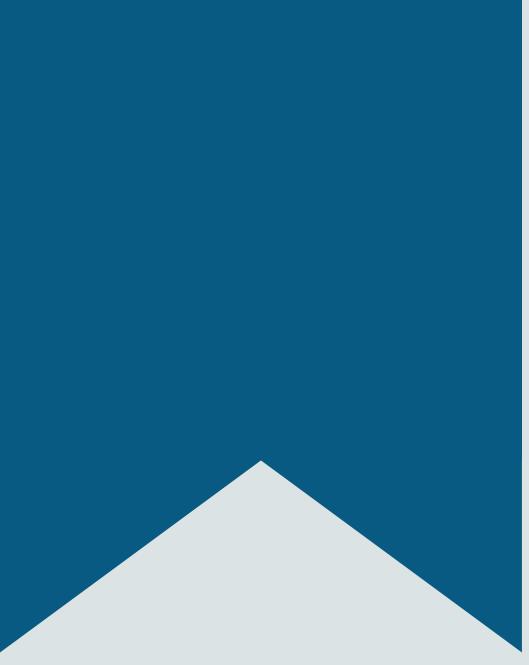


Dashboard Creation: Final Screen

Post configuration, the dashboard will look like this:

The screenshot shows the Azure DevOps interface with the project 'SampleProject-DevOps' selected. The dashboard for 'Azure-PipeLine-Cl' is displayed. On the left, a sidebar lists 'Overview', 'Summary', 'Dashboards' (which is selected), 'Wiki', 'Boards', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main area shows a summary bar chart with green bars and a red bar, labeled 'Monday'. Below it, a section titled 'Work assigned to you' shows counts for Epic (1), Issue (1), and Task (1). A table lists three work items: ID 162 (Epic 1), ID 163 (Issue 1), and ID 164 (Task 1), all in 'To Do' state. A 'New Work Item' form is visible on the right.

Now, users can see all the relevant information on the dashboard itself. This helps in better co-ordination and reduced effort.



Integrate and Configure Ticketing Systems

Ticketing System

The ticketing system is used for creating and updating work-items and incident management

The ITSM tools are used to:

- Create or update work-items in the ITSM tools based on Azure alerts
- Pull incident and change request data from ITSM tools into Azure Log Analytics

ITSM Tools

ServiceNow

It is a software as a service (SaaS) product for technical management support

System Center Service Manager

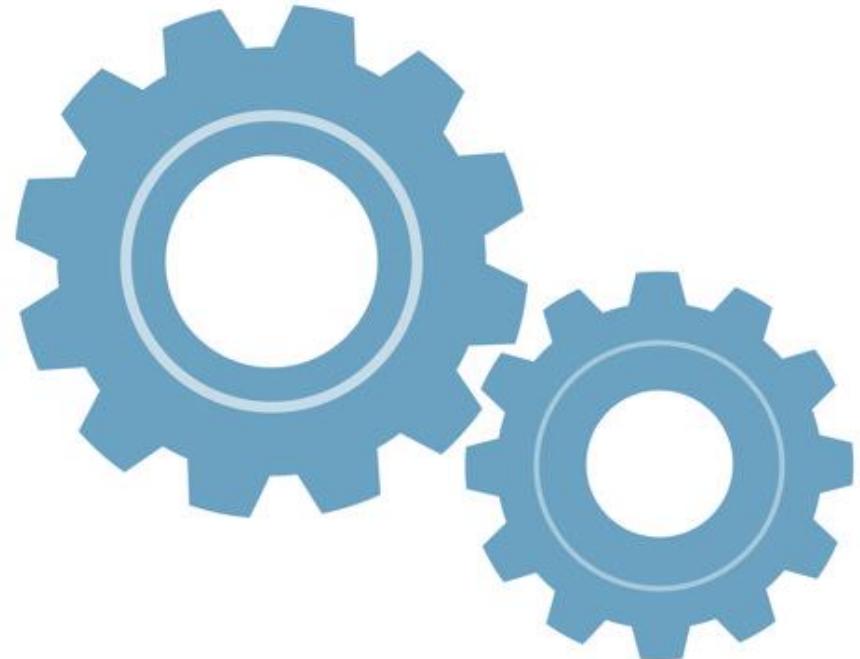
It offers an integrated interface to automate and adopt the IT service management best practices

Cherwell

It helps IT teams implement, automate, and upgrade service and support processes

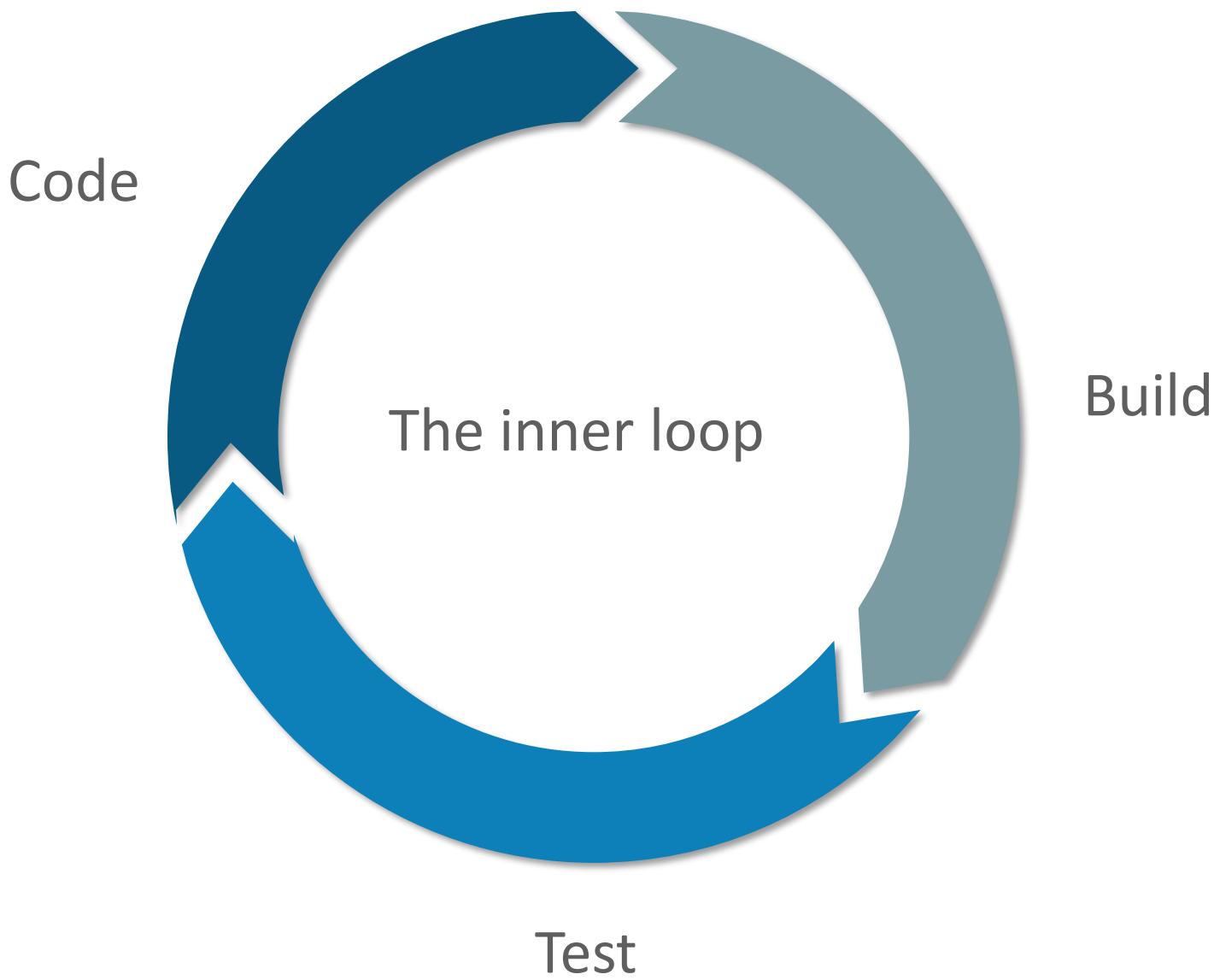
Provance

It provides a cost-effective IT Service and IT Asset Management



The Inner Loop

It is the iterative process that developers perform when they write, build, and debug the code.



Inner Loop Optimization

Developers can optimize the inner loop through the below-mentioned points:

1

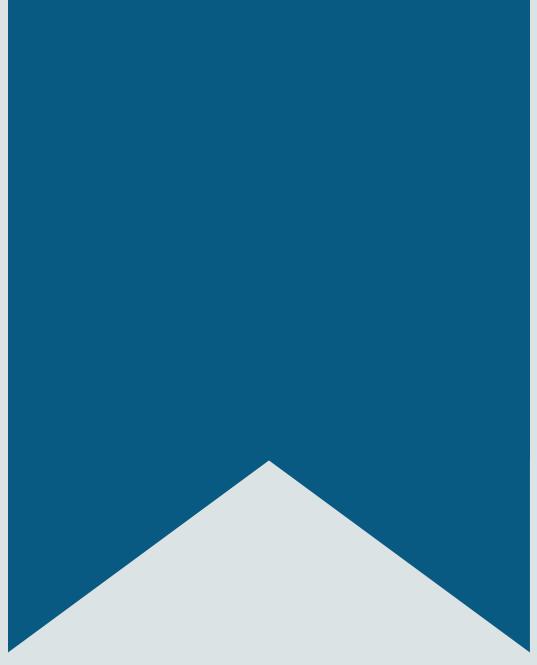
In case of any new change, only build & test the new change/code

2

Implement caching. This will reduce the amount of code, and thus build process can be improved with less time

3

If possible, break the application into smaller components



Design Practices for Measuring End-User Satisfaction

Parameters Affecting User Experience

For measuring end-user satisfaction, it is important to measure the parameters which affect the satisfaction of end-users. They are:

Request rate of application

Response time of application

Failure rate while responding to request, e.g., timeout

Performance issue through external factors like low internet speed

Application exception

High utilization of system resources

Application Insights

- Azure provides **Application Insights**, which monitors all these parameters, and it is added to Azure release pipeline
- It gets the telemetry data before calculating these parameters



Incorporating Application Insights: Step 1

Go to the Azure Pipeline of the project

Click on Releases and Create Release

The screenshot shows the Azure DevOps interface for a project named "SampleProject-DevOps". The left sidebar has "Azure-PipeLine-Cl" selected under "Pipelines". The main area is titled "New release pipeline" with the sub-header "No releases found". It features a search bar, a "New" button, and tabs for "Releases", "Deployments", and "Analytics". A "Create a release" button is at the bottom right. The page includes a cartoon illustration of a person and a dog launching a rocket.

Incorporating Application Insights: Step 2

Search for Application Insights under Add task and click on "Get it free," as shown here:

All pipelines > New release pipeline

Pipeline Tasks Variables Retention Options History

Stage 1 Deployment process

Run on agent Run on agent

Deploy Azure App Service Azure App Service deploy

Add tasks Refresh Marketplace

Release Annotations for Azure Application Insights
See your releases marked on your Application Insights timelines.
by Microsoft | 5,275 installs

Get it free Learn more

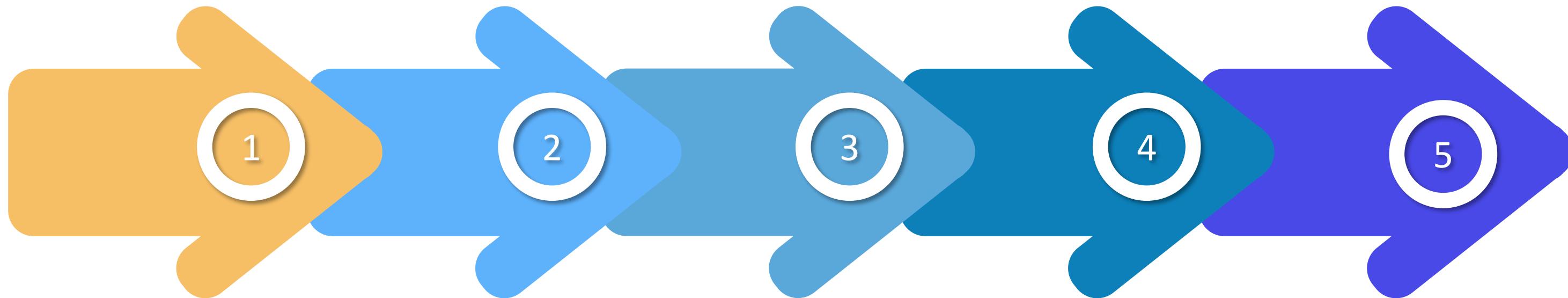
Azure Cost Insights

Incorporating Application Insights: Step 3

The control will navigate to Azure marketplace webpage

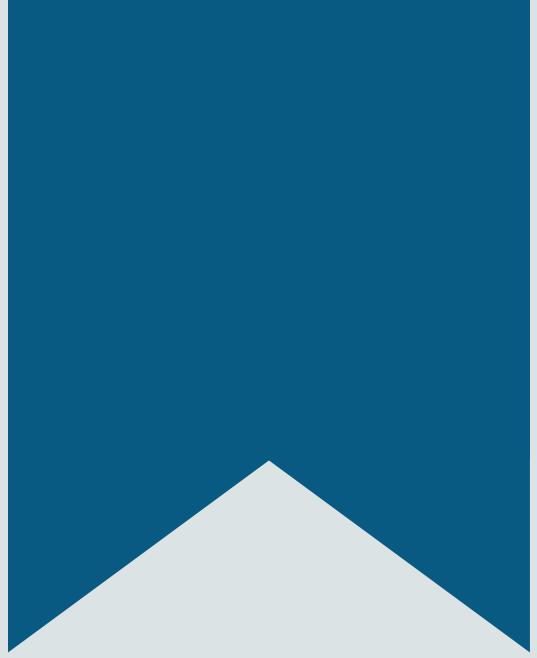
Once the user moves to Project and goes to the same release process, you can now search "Release annotation"

This will add Application Insights as part of the application deployment



Select the project, install it, and click on "Proceed to the organization"

Select this and click on Add. Put the required parameters of this task



Design Processes for Capturing and Analyzing user Feedback

Feedback

- Feedback is an important part of the design and development process from various stakeholders
- Feedback is sought from stakeholders for user stories or features defined
- Users can request feedback for any of the stakeholders

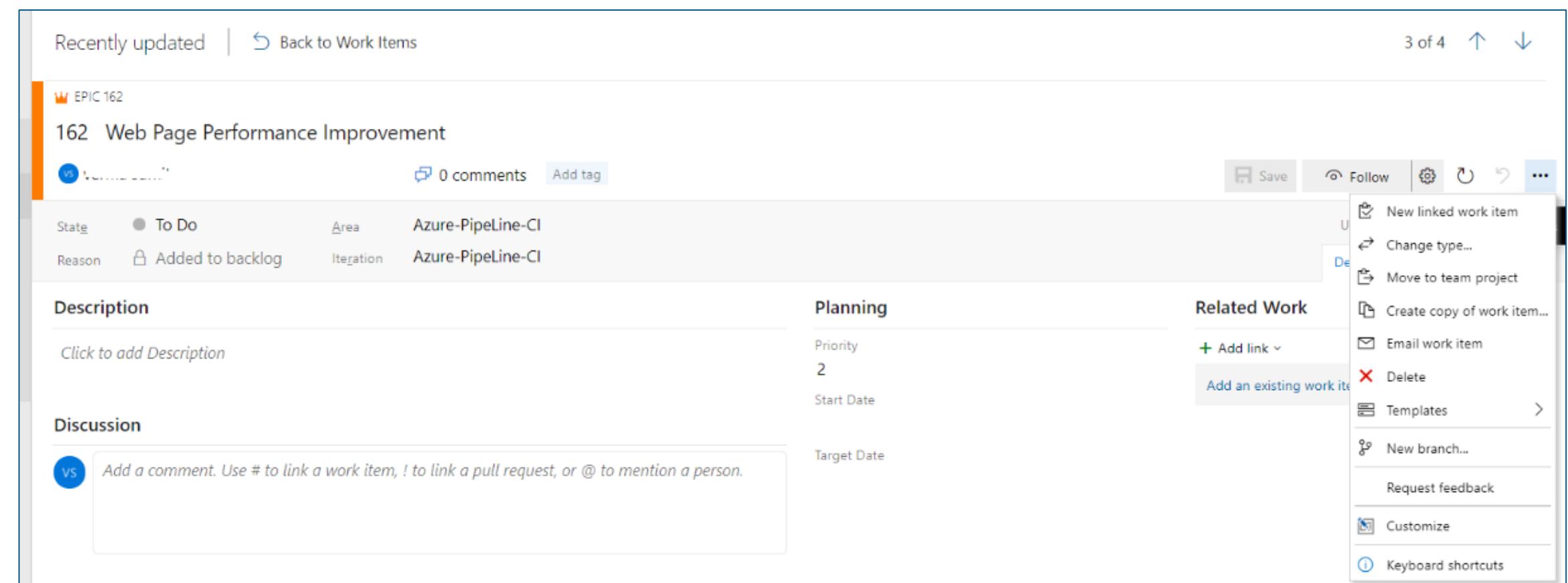


Capturing User Feedback

Go to the work item
of the project for
which feedback is
required

Click on the "..." as
shown here

Click on Request
feedback



Capturing User Feedback (Cont.)

After this, a window will open where the user can write about the feedback and specify the stakeholder

Request feedback

To

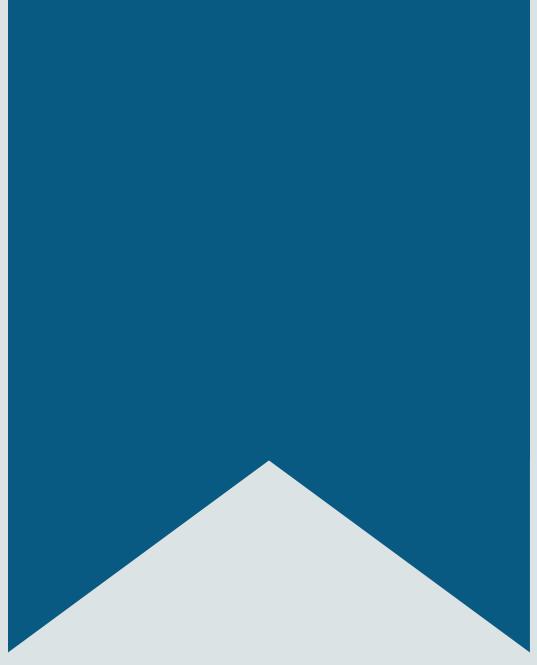
Subject

Feedback on

Instructions

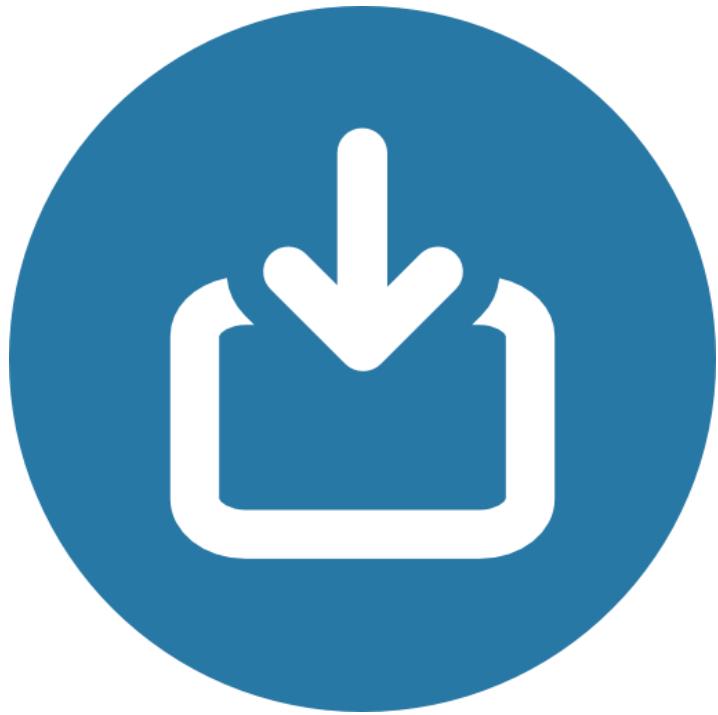
0/1024 characters

Stakeholders will get mail for feedback



Design Process for Automating Application Analytics

Analytics Widget

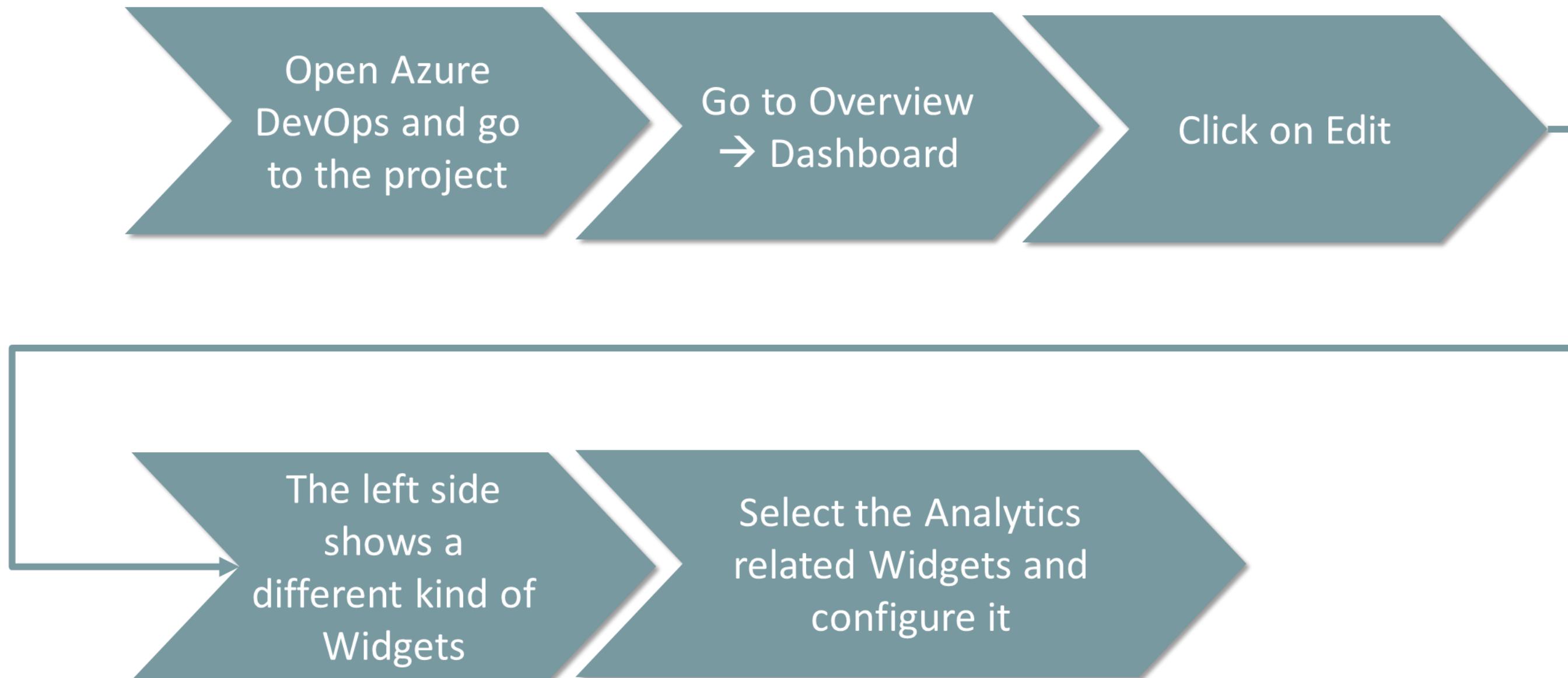


Application analytics can be designed and viewed dynamically through Analytics Widget

There are different Analytics widgets offered by Azure DevOps like Burndown Widgets, Lead Time, Velocity

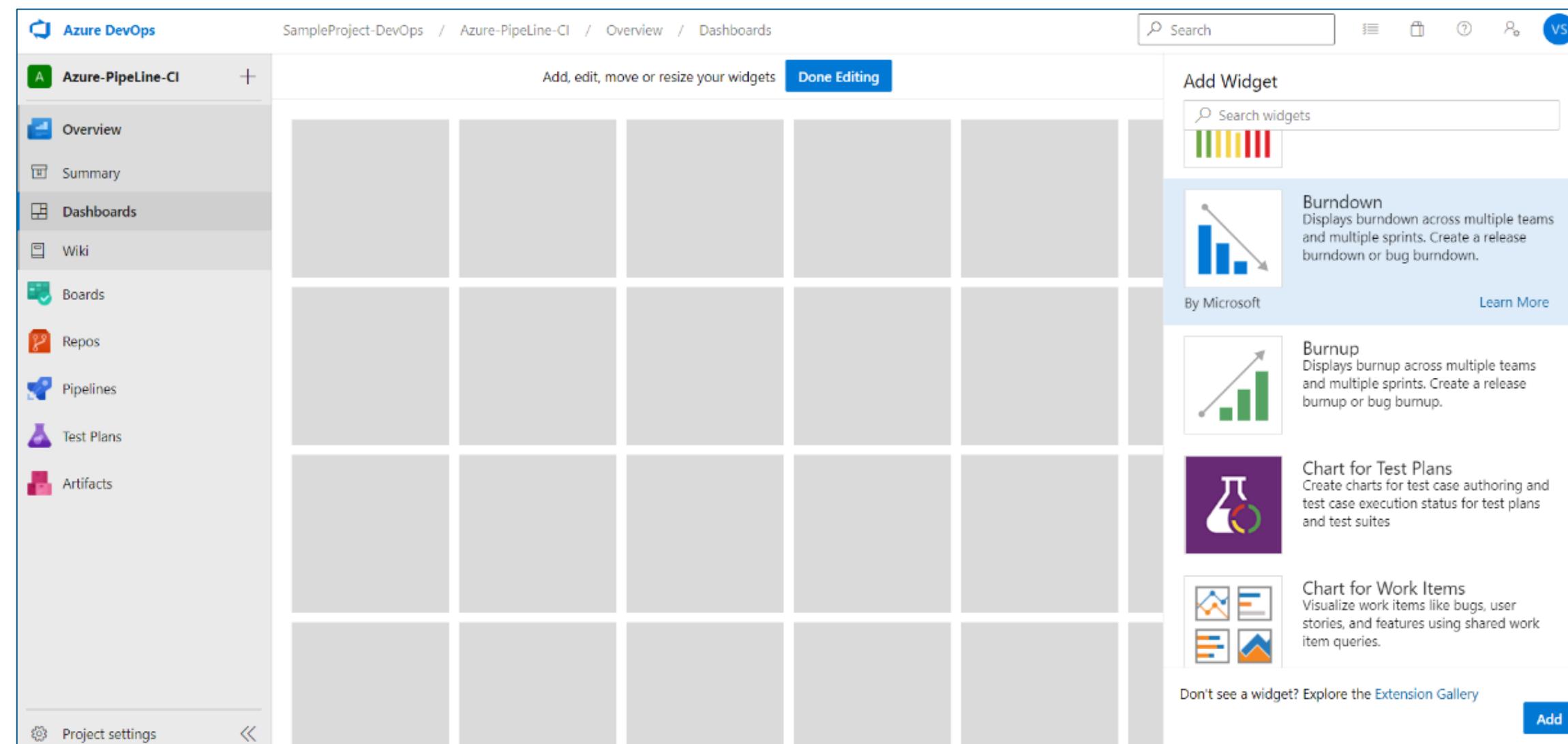
These widgets help visualize the analytics data on the dashboard dynamically

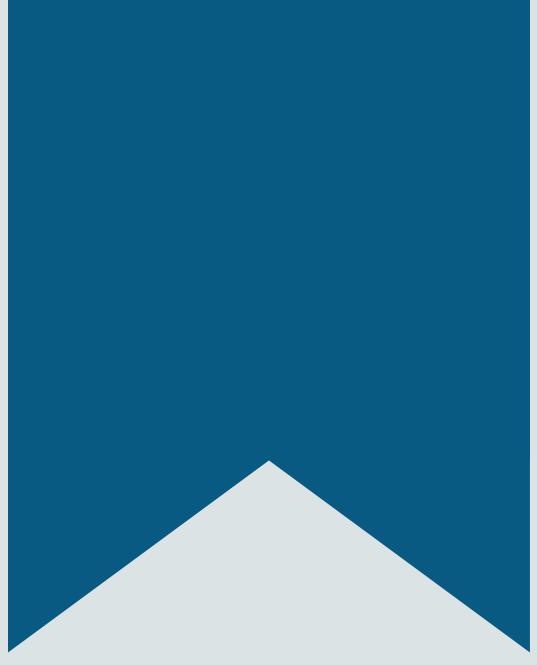
Steps to Configure Widgets



Widget with the Analytics

By performing the discussed steps, the widget containing the analytics will be shown on the dashboard.





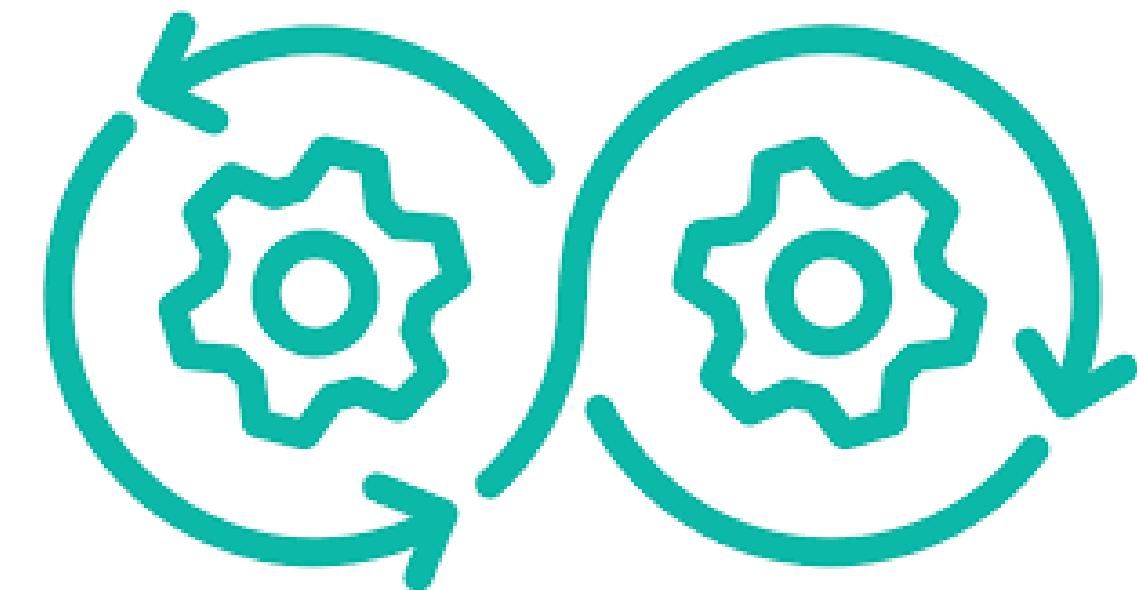
Site Reliability Engineering

Site Reliability Engineering: Overview

Site reliability engineering (SRE) is a practice devoted to helping organizations achieve the right level of reliability in their system, product, and services

The objective is to achieve reliability in the software that is deployed in production

SRE engineers focus on improving the reliability of any software or services so that customer doesn't face any issue



Site Reliability Engineer: Responsibilities

Monitor and analyze the performance of applications



Handle on-call and emergency support



Ensure good logging and diagnostics for the applications



Create/manage runbooks for operations



Help triage escalated support tickets



Work on requests for features and defects



Contribute to the overall roadmap of products



Perform live site reviews and collect feedback



Site Reliability Engineer: Skills

Application monitoring and diagnostics



Knowledge of Application Insights



Application logging



Exception handling

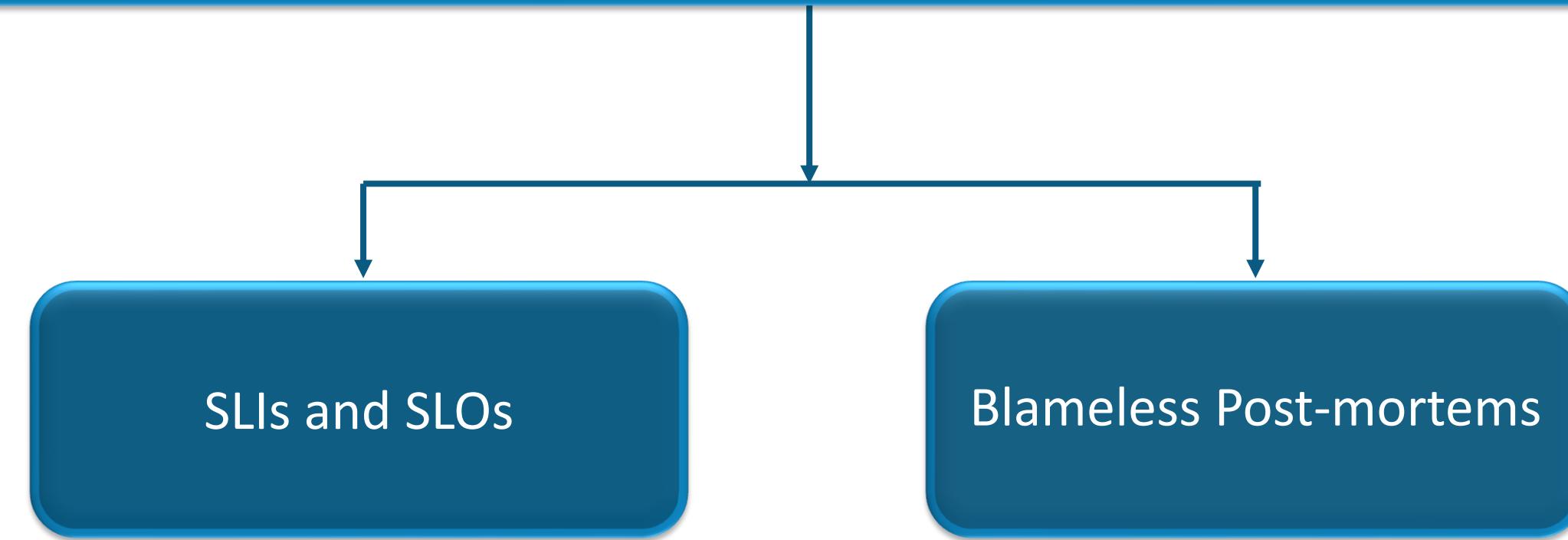


In-depth knowledge of product/application



SRE: Key Principles and Practices

For SRE, there are two principles defined. These are also known as the virtuous cycle:



SLIs and SLOs

- SLI (Service Level Indicator) is an indicator of service health, helping measure the performance of a system
- SLO (Service Level Objective), on the other hand, is an indicator of the reliability level that we can expect



Blameless Post-Mortems

- Blameless Post-Mortems help comprehend why a certain system failed through a deep analysis
- It can be due to myriad reasons
- This analysis is required to revamp the process so that failure doesn't occur in the future





Establish a Baseline by Analyzing Telemetry

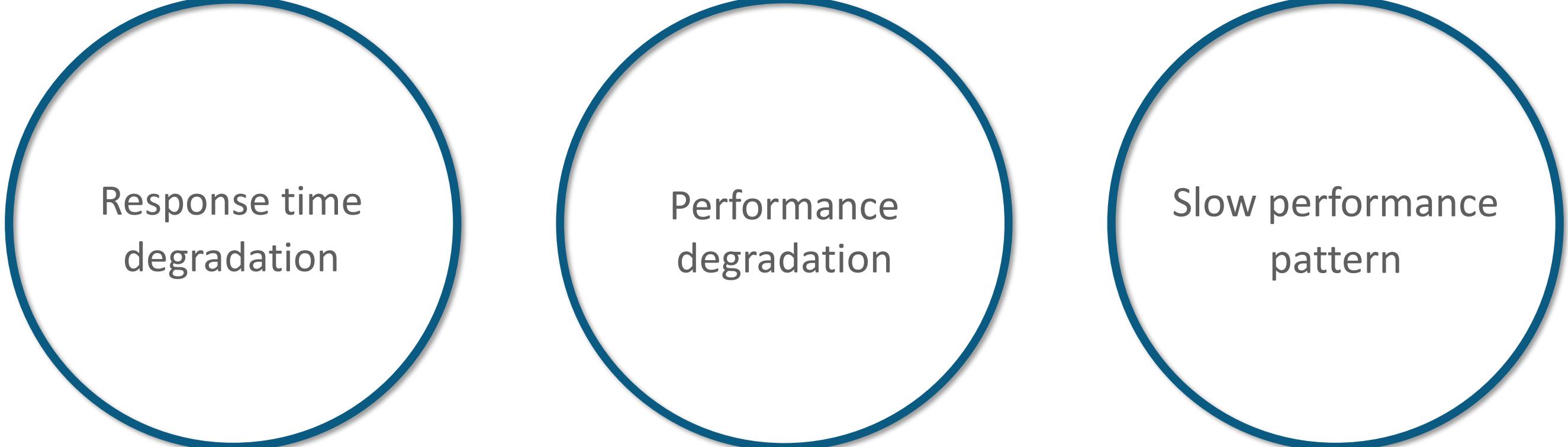
Telemetry of an Application

The telemetry of the application is analyzed through **Application Insights**.



Issues that Trigger Application Insights

Application Insights sends the notifications if it detects any of the following:



- Response time degradation

- Performance degradation

- Slow performance pattern

For notification, Application Insights offers a unique feature known as Smart Detection.

Viewing Smart Detections

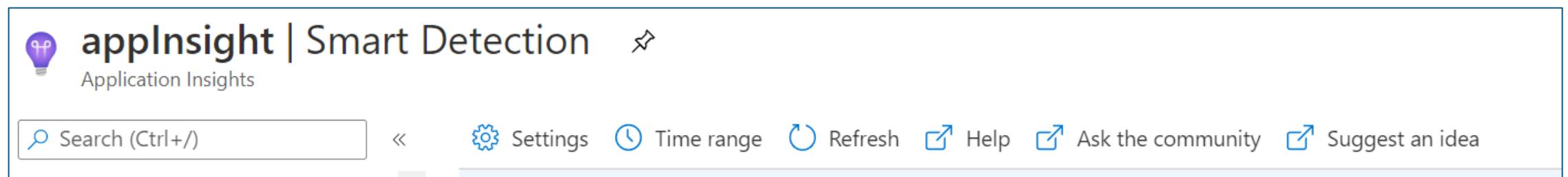
Smart detections can be viewed from the smart detection blade of Application Insights or through the mails sent

The screenshot shows the 'Smart Detection' blade in the Application Insights blade. The sidebar on the left lists various monitoring and diagnostic features. The 'Smart Detection' item is highlighted with a red box. The main area displays four detected issues:

- Abnormal rise in exception volume (preview)**
When: 7/28 5:30 AM - 7/29 5:29 AM
What: 450% increase in 'System.Web.HttpException' volume compared to the previous 7 days
Note: Data provided by 1 user was potentially compromised due to this insecure data transmission
- Insecure form data transmission detected (preview)**
When: 7/9 5:30 AM - 7/10 5:29 AM
What: 2 operations or forms in your application submit data to insecure (non-HTTPS) URLs
Note: Data provided by 1 user was potentially compromised due to this insecure data transmission
- Potentially insecure URL access detected (preview)**
When: 7/9 5:30 AM - 7/10 5:29 AM
What: 2 URLs were accessed by both HTTP and HTTPS protocols
Note: 4 users accessed multiple URLs using HTTP instead of HTTPS
- Abnormal rise in exception volume (preview)**
When: 6/29 5:30 AM - 6/30 5:29 AM
What: 575% increase in 'System.Web.HttpException' volume compared to the previous 7 days
- Abnormal rise in exception volume (preview)**
When: 6/14 5:30 AM - 6/15 5:29 AM
What: Significant increase in 'Object doesn't support property or method 'Symbol.iterator'' volume compared to the previous 7 days

Smart Detection Settings

Smart detection setting can be specified by clicking on the Settings tab of the Smart detection page



Smart Detection Settings (Cont.)

Select any of the rules to configure it and specify mail id for notification

ENAB...	NAME	INFO	SEVERITY
✓	Slow page load time	ⓘ	Information
✓	Slow server response time	ⓘ	Information
✓	Long dependency duration	ⓘ	Information
✓	Degradation in server response time	ⓘ	Information
✓	Degradation in dependency duration	ⓘ	Information
✓	Degradation in trace severity ratio (preview)	ⓘ	Information
✓	Abnormal rise in exception volume (preview)	ⓘ	Information
✓	Potential memory leak detected (preview)	ⓘ	Information
✓	Potential security issue detected (preview)	ⓘ	Information
✓	Abnormal rise in daily data volume (preview)	ⓘ	Information

Steps to Improve Performance

Triaging is to find the level of importance the issue holds

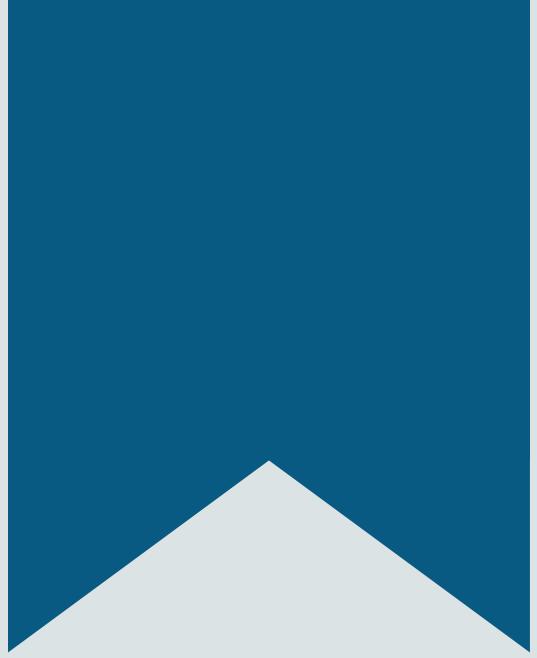
Triage

The issue should be diagnosed, and root issues should be found for this issue

Diagnose

Address the root issues to resolve the performance issue

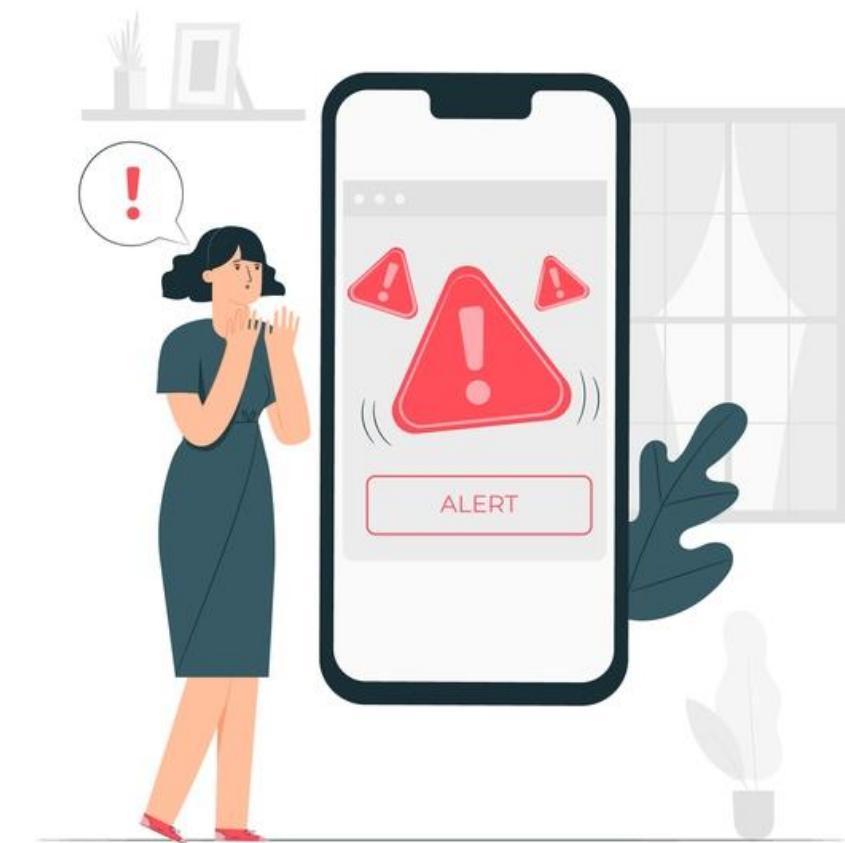
Improve



Perform Tuning to Reduce Meaningless or Non-Actionable Alerts

Ticket System Alerts

- Ticket System Alerts find out meaningless or non-actionable tickets
- If the alerts must be ignored, the rule defining the alerts should be removed from Application Insights



Reducing Meaningless/Non-Actionable Alerts

The ways to reduce meaningless or non-actionable alerts are:



Bug or ticket
tracking system



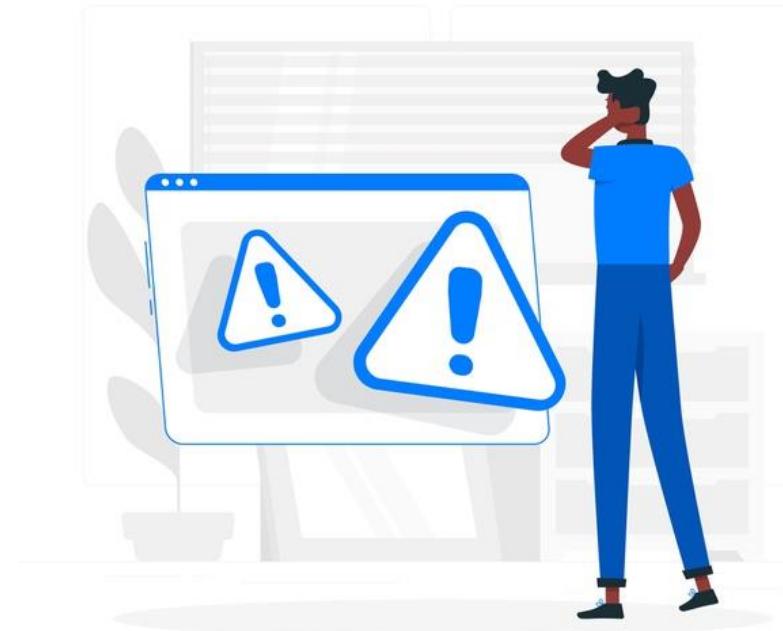
A daily report of
bugs



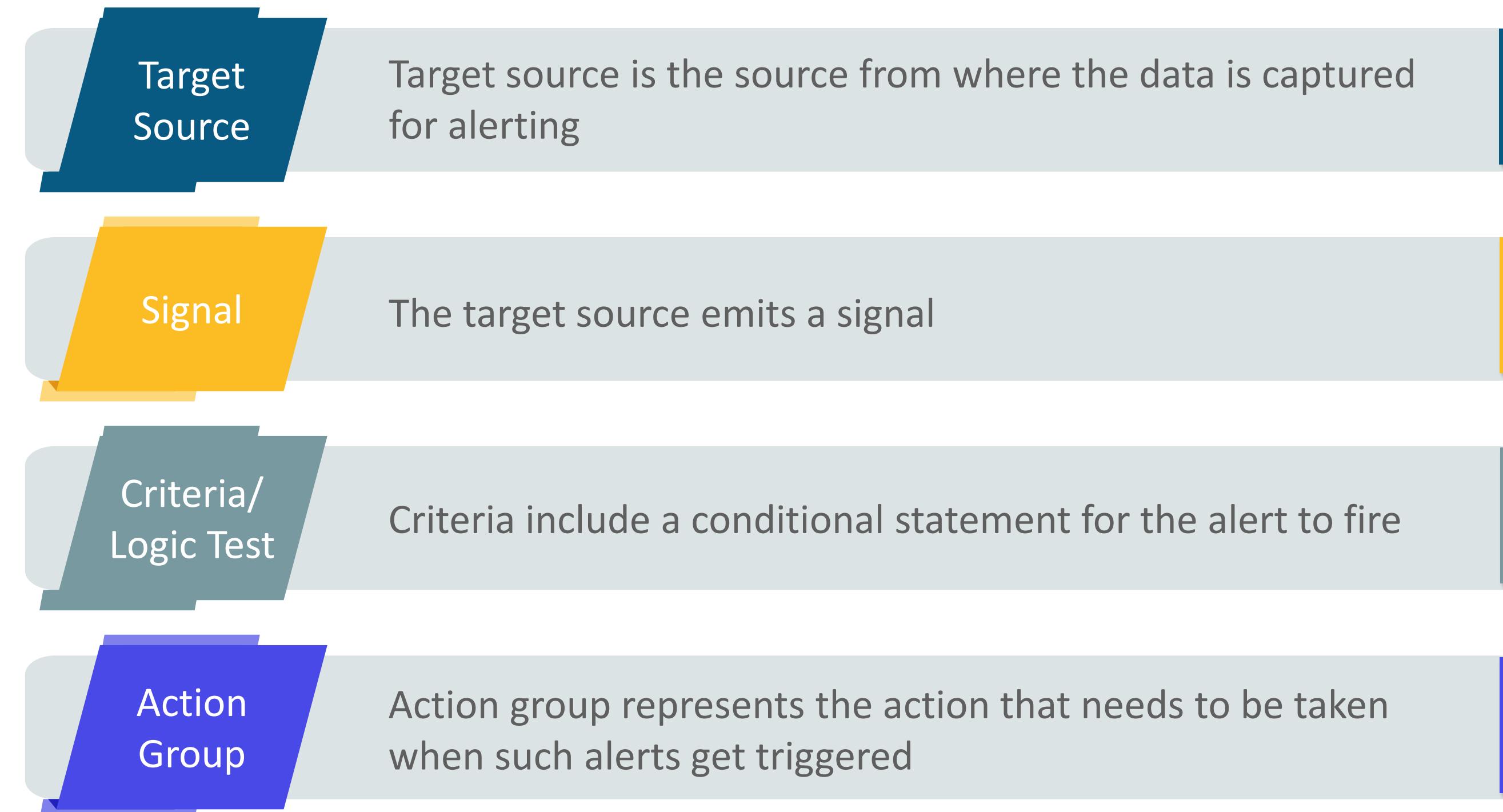
Tracking alerts

Analyzing Alerts to Establish a Baseline

- Alerts are triggered while analyzing the telemetry data by Application Insights
- The alert rule captures the target and criteria for alerting
- The alert rule can be in an enabled or a disabled state



Key Attributes of Alert Rule



Creating Action Group: Step 1

An action group has many options and can be created from an alert screen, as shown here:

The screenshot shows the 'Create action group' page in the Azure portal. The top navigation bar includes 'Home > Alerts > Manage actions > Create action group'. Below the navigation is a tabs menu: Basics, Notifications, Actions (which is underlined), Tags, and Review + create. The main content area is titled 'Actions' and contains the following text: 'Configure the method in which actions are performed when the action group triggers. Select action types, fill out associated details, and add a unique description. This step is optional.' A table lists two actions: 'ITSM' (Name: 'Create alert in ITSM system', Selected: 'Alert') and 'Webhook' (Name: 'Run remediation', Selected: 'URI'). To the right of each row are edit and delete icons. A dropdown menu is open, showing additional action types: Automation Runbook, Azure Function, ITSM, Logic App, Secure Webhook, and Webhook. At the bottom of the page are buttons for 'Review + create', 'Previous', and 'Next: Tags >'.

Action type ⓘ	Name ⓘ	Selected ⓘ
ITSM	Create alert in ITSM system	Alert ⓘ
Webhook	Run remediation	URI ⓘ

Automation Runbook
Azure Function
ITSM
Logic App
Secure Webhook
Webhook

Review + create Previous Next: Tags >

Creating Action Group: Step 2

Notifications tab
represents the alert to be
received through email, as
shown here:

Create action group

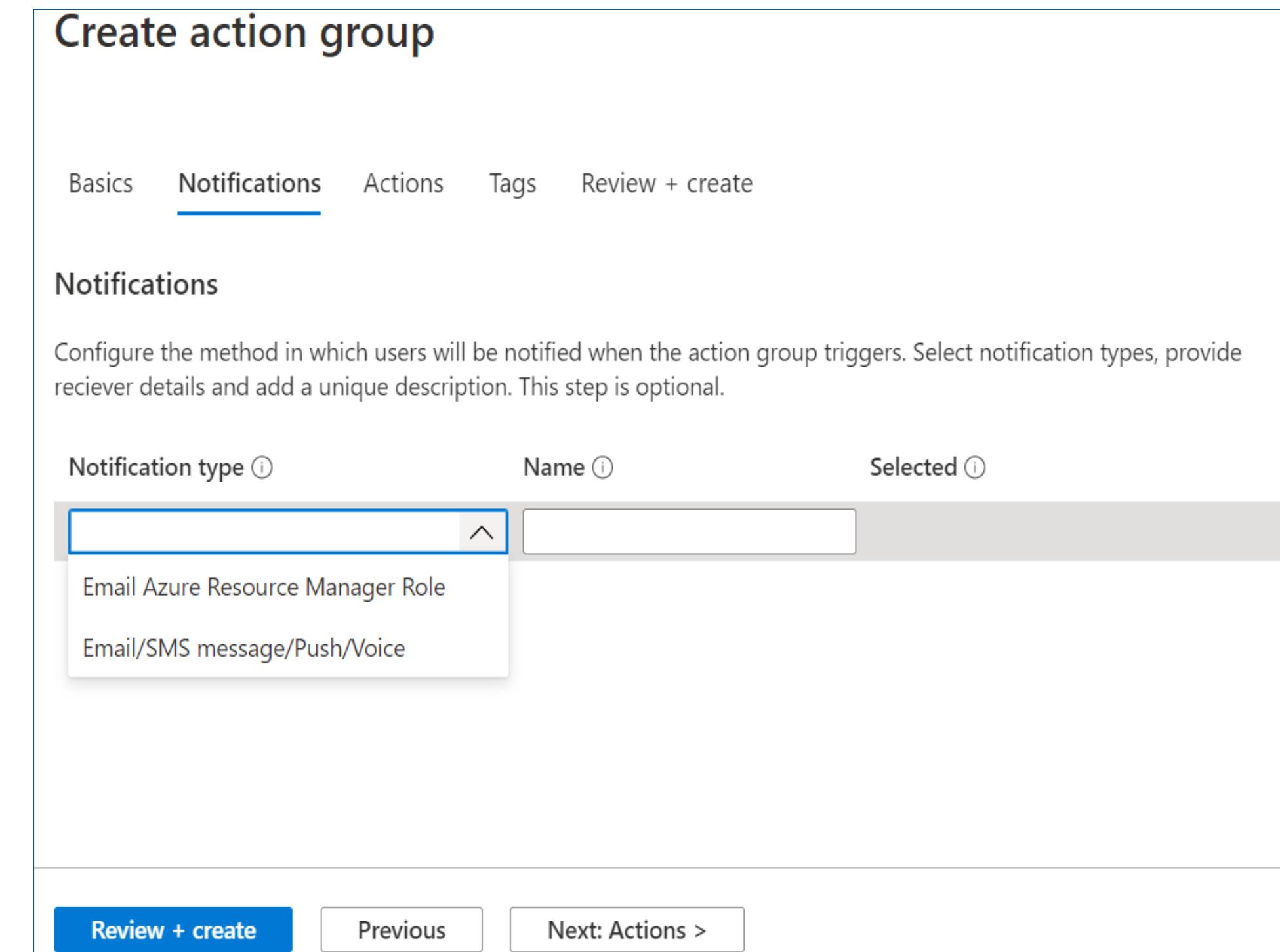
Basics Notifications Actions Tags Review + create

Notifications

Configure the method in which users will be notified when the action group triggers. Select notification types, provide receiver details and add a unique description. This step is optional.

Notification type ⓘ	Name ⓘ	Selected ⓘ
Email Azure Resource Manager Role		
Email/SMS message/Push/Voice		

Review + create **Previous** **Next: Actions >**



Creating Alert Rule: Step 1

Click on Alerts on Application insights. Then, click on "New Alert Rule"

The screenshot shows the Azure Application Insights Alerts blade. On the left, there's a sidebar with options: Search (Ctrl+ /), Search, Availability, Failures, Performance, Troubleshooting guides (previous), Monitoring (which is selected and highlighted in grey), Alerts (selected and highlighted in blue), Metrics, Logs, Workbooks, Usage, Users, and Sessions. The main area has a header with 'New alert rule', 'Manage alert rules', 'Manage actions', 'View classic alerts', 'Refresh', and 'Provide feedback'. Below the header, it says 'Don't see a subscription? Open Directory + Subscription settings'. It shows 'Subscription * Pago por uso', 'Resource group Insight', 'Resource applInsight', and 'Time range Past 24 hours'. A message 'Selected subscriptions > Insight > applInsight' is displayed. To the right, a chart titled 'All is good! You have no alerts.' shows a pie chart with a large blue slice and a bar chart with three bars.

Creating Alert Rule: Step 2

Click on Select condition, and a new window will pop up describing all the alert parameters

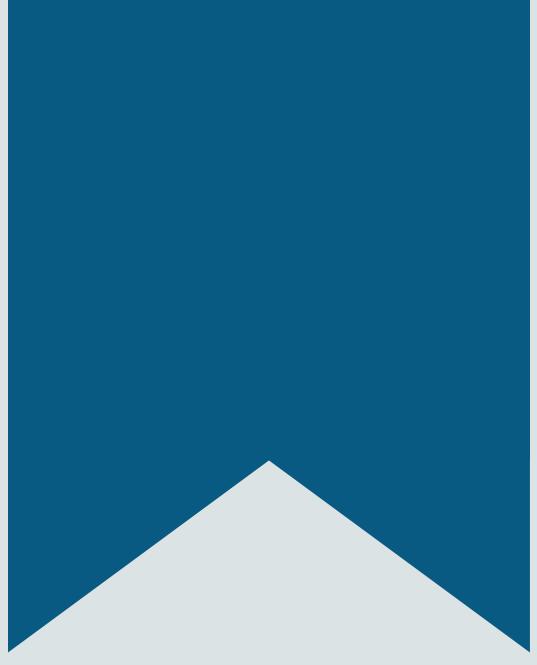
Select the signal and define the threshold values that define the baseline

Configure signal logic

Displaying 1 - 20 signals out of total 41 signals

Signal name	Signal type	Monitor service
Custom log search	Log	Application Insights
Availability	Metric	Platform
Availability tests	Metric	Platform
Availability test duration	Metric	Platform
Page load network connect time	Metric	Platform
Client processing time	Metric	Platform
Receiving response time	Metric	Platform
Send request time	Metric	Platform
Browser page load time	Metric	Platform
Dependency calls	Metric	Platform

Done

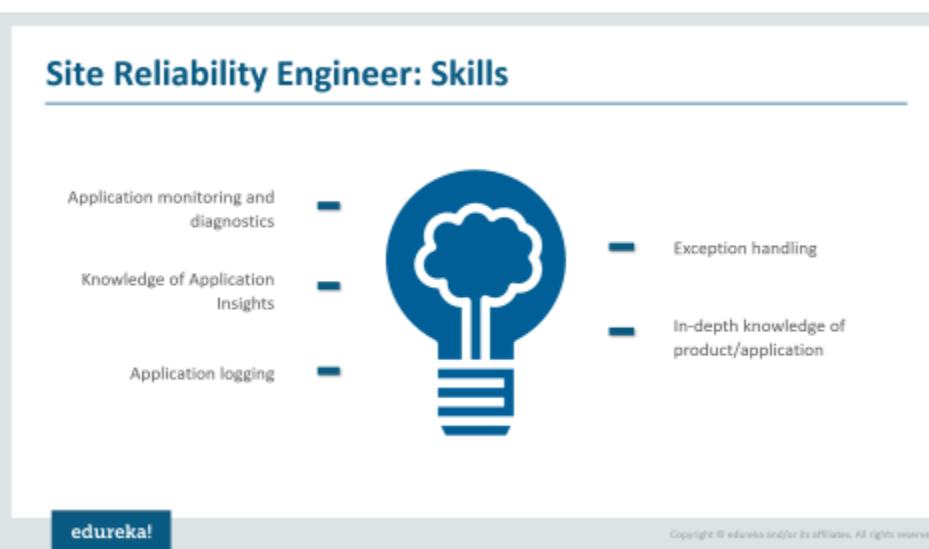
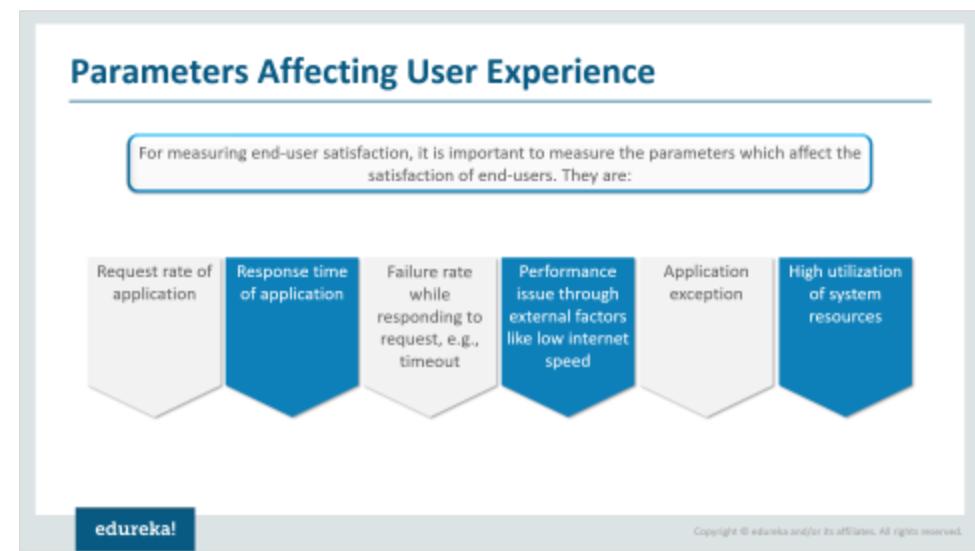
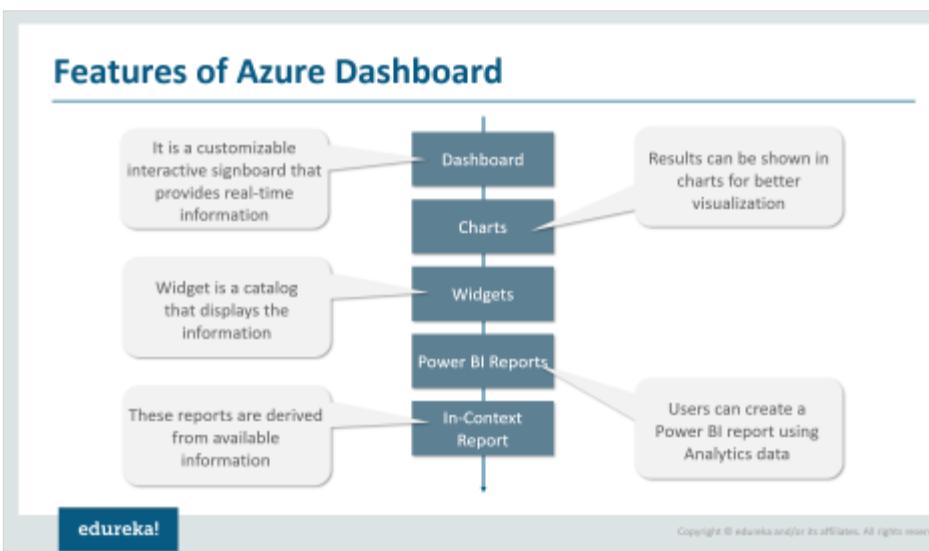
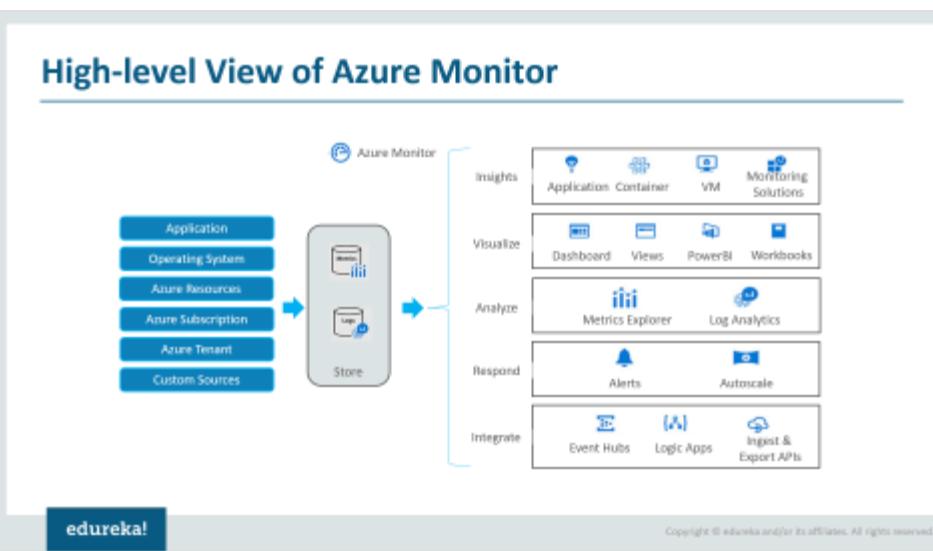


Demo: Monitor Application Performance



Demo: Integrate between Azure DevOps and Team

Summary



Questions

FEEDBACK



Survey



Ideas



Ratings



Comments



Suggestions



Likes

Thank You



For more information please visit our website
www.edureka.co