

# The 4 Use Cases of **SlackOps** in DevOps

[NEXT](#)

# SlackOps It's the practice of using Slack as a central hub to manage operations

*Not just for communication but to optimize operational efficiency and foster collaboration.*



The screenshot shows the Slack desktop application interface. On the left, the sidebar displays the 'Test Team' channel, listing various channels like '# cats', '# emoji-cat', and '# general'. The '# project-blueprints' channel is currently selected and highlighted in blue. In the main workspace, a message from 'girlie\_mac' at 2:05 PM is shown, stating: 'renamed the channel from "blueprints" to "project-blueprints"'. Below it, a message from 'channel-naming-convention' at 2:05 PM says: 'No matching channel template found for project. Do you want to add it or use an existing template?'. At the bottom of the screen, there is a message input field with placeholder text 'Message #test-blueprints' and a rich text editor toolbar.





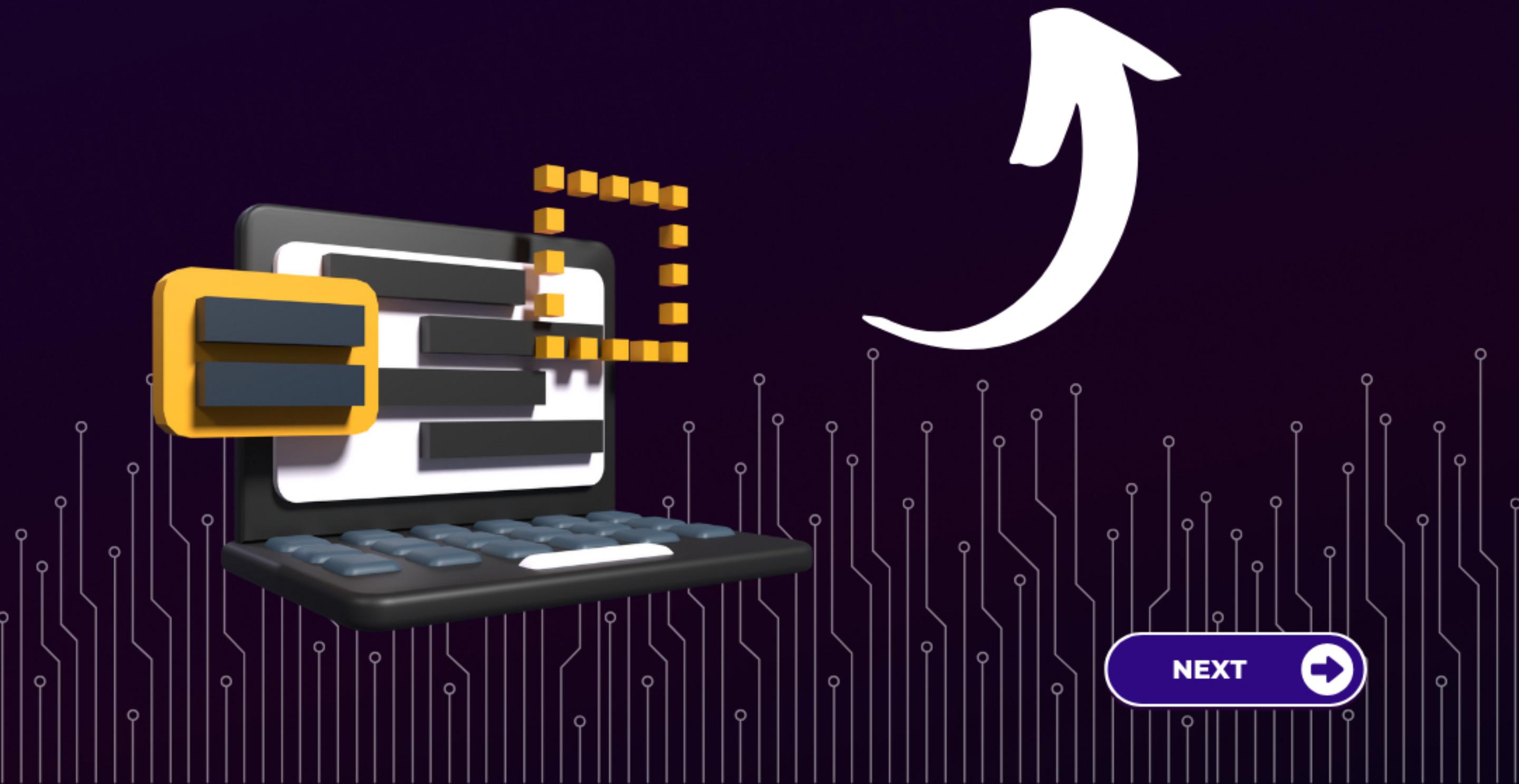
Successful SlackOps adoption starts with understanding the DevOps workflow and addressing specific organizational pain points.



# 1 Monitoring

SlackOps can integrate with DevOps tools for real-time updates on system health.

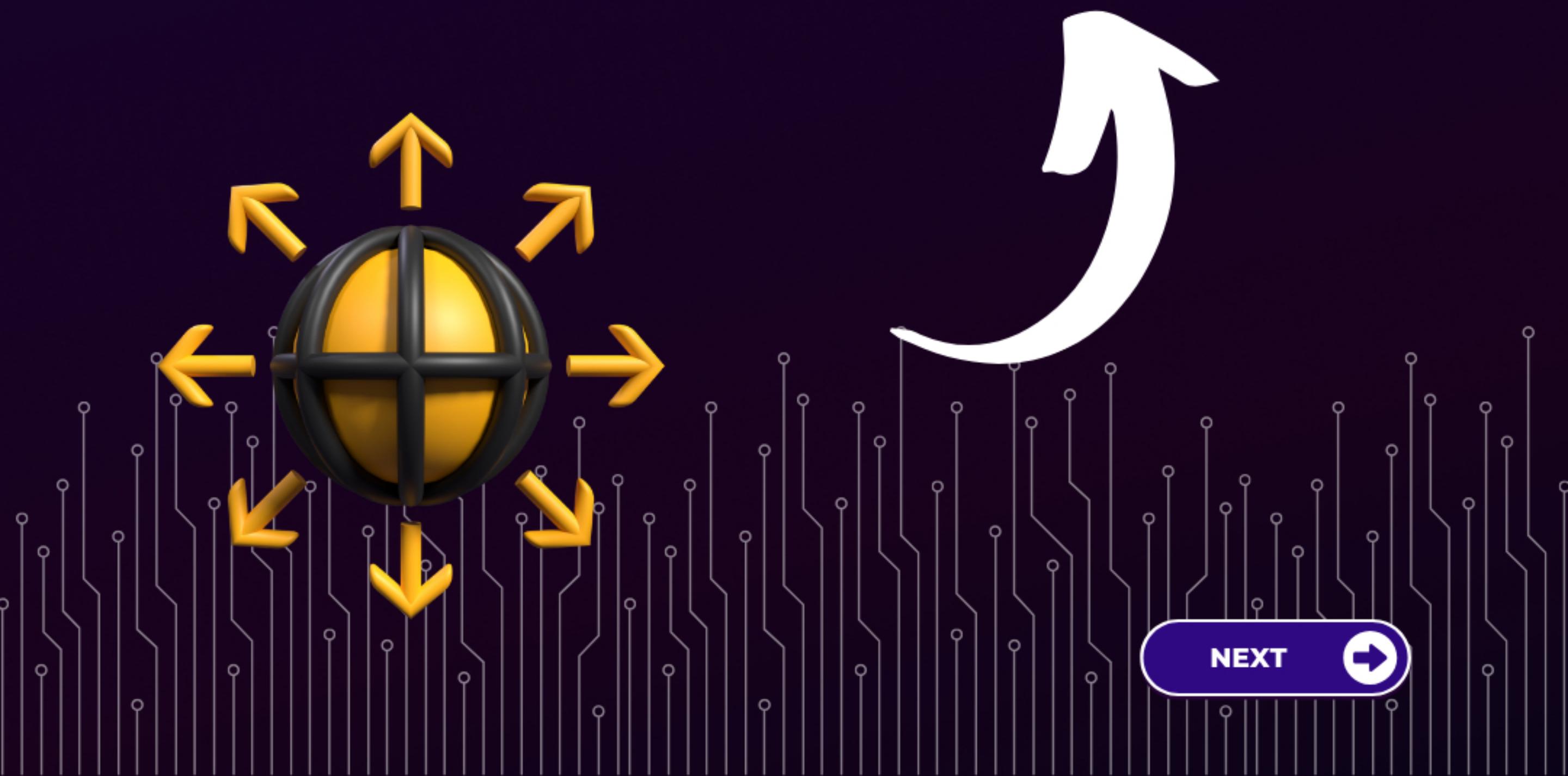
*performance metrics, and incidents leading to rapid issue identification and resolution.*



## 2 Deployment management

Integrating Slack with CI/CD pipelines provides automated updates about deployment status, code changes, and test results.

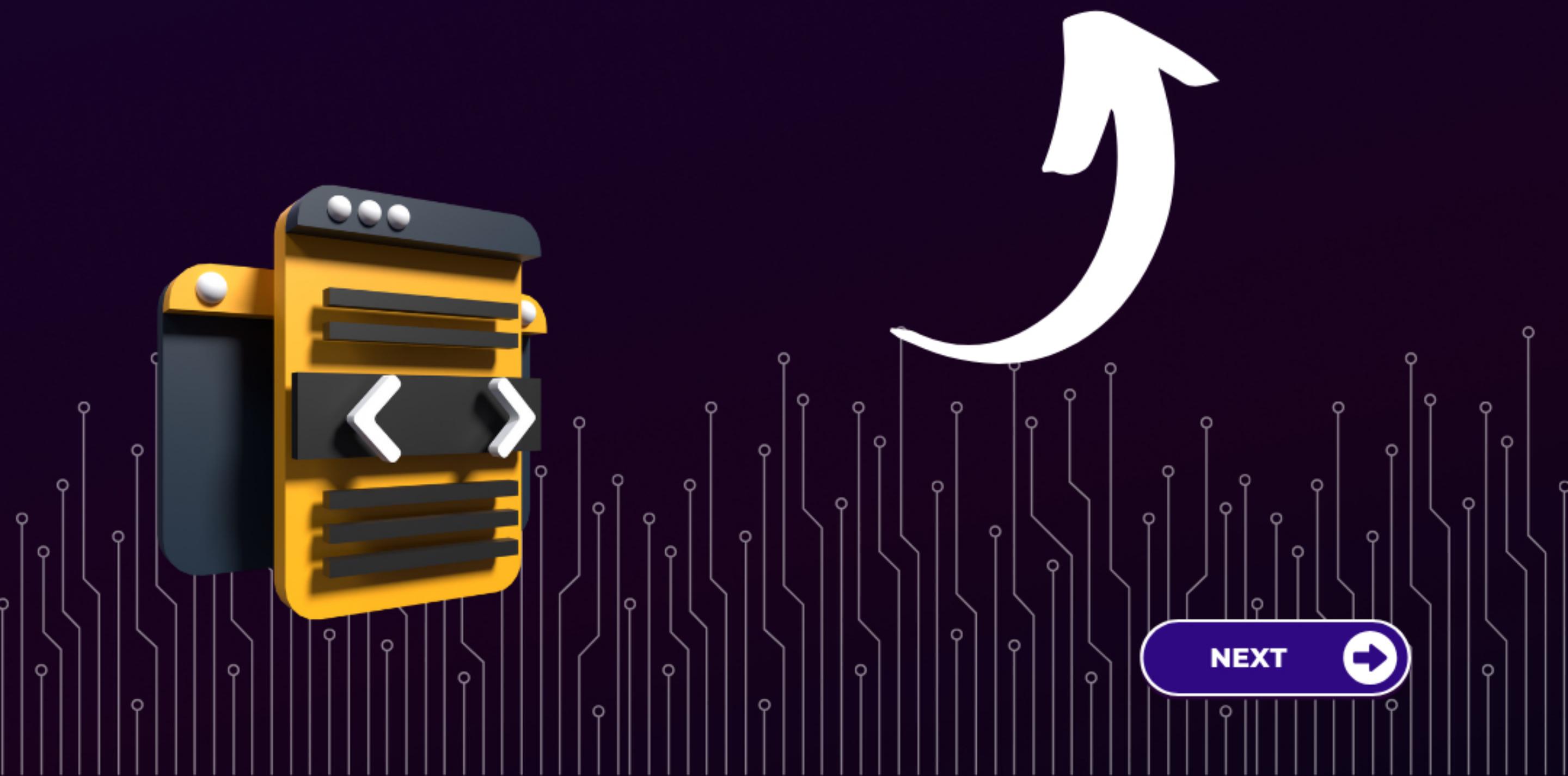
*promoting transparency and fast response to potential failures.*



# 3 Code changes

Notifications of new code pushed to the repository, automated test results can be sent to Slack channels.

*Enabling immediate review and feedback.*

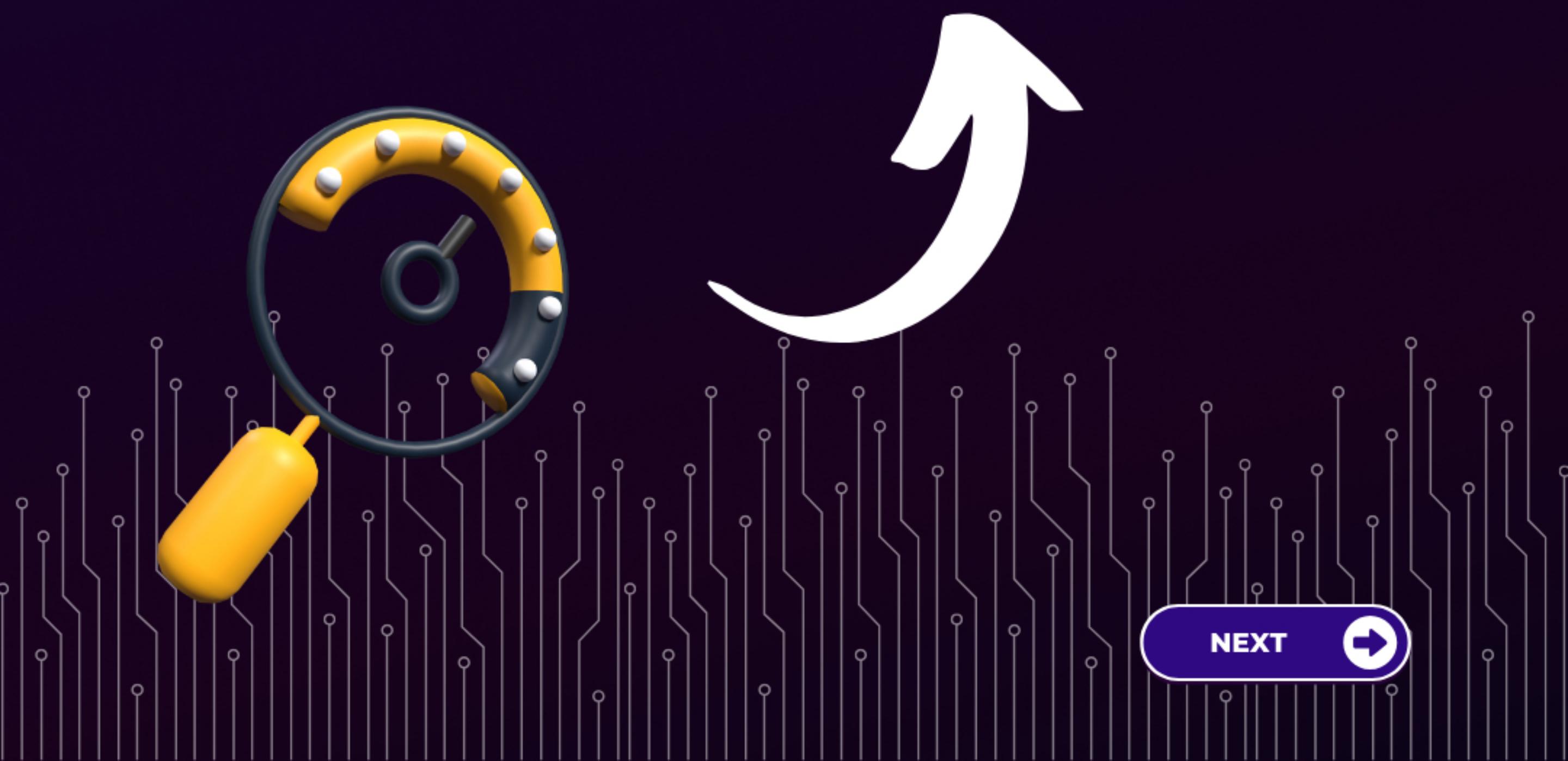


## 4

# Incident management

With dedicated channels and integrations with management tools.

*SlackOps centralizes communication during critical situations, reducing downtime and improving reliability.*





**Kubiya** is the missing ingredient in SlackOps, where users can take advantage of almost an unlimited array of operational self-service functions without overburdening the devops engineers.

By leveraging LLM capabilities throughout the entire stack, **Kubiya** an intelligent agent that lives within chat engines allows its users to interact in natural free form language and express their intent without requiring prior context.



 **Kubiya Next Gen** APP 10 days ago  
Great choice! You've selected the action `get_application` which requires some parameters. Let's collect them:

1. `appNamespace` (type: string) - This is the namespace of the application you want to get.
2. `name` (type: string) - This is the name of the application.
3. `project` (type: array) - This is the project that the application belongs to. You can provide multiple projects if needed.

Could you please provide the `appNamespace` and `name` of the application? Also, let me know the `project` it belongs to. If there are multiple projects, you can list them all. 

 **A** **Andrew Kuang** 10 days ago  
`@Kubiya Next Gen` appNamespace: dev, name: helm-webapp, project: default

 **Kubiya Next Gen** APP 10 days ago  
✓ Got it! Here's what I have:  

- `appNamespace`: dev
- `name`: helm-webapp
- `project`: default

Let's confirm this. Here's the JSON object for the `get_application` action:

```
json
{
  "appNamespace": "dev",
  "name": "helm-webapp",
  "project": ["default"]
}
```

