Information on run game days + FIS

Gamedays

AWS GameDay is a simulation-based learning experience that helps teams learn how to operate in a high-stress, real-world environment. GameDays are designed to test and improve a team's ability to respond to incidents and recover from failures. In the context of AWS serverless infrastructure, GameDays can be used to test your disaster recovery plan and ensure that your serverless applications are resilient to potential disasters.

Here are some key steps for planning and executing a recovery-focused AWS GameDay for serverless infrastructure:

1. Define the objectives: Define the objectives of your GameDay. This could include testing your disaster recovery plan, identifying areas for improvement in your recovery processes, or improving your team's ability to respond to incidents.
2. Define the scenario: Define the scenario for your GameDay. This should be a realistic disaster scenario that your team could potentially face. **For example, you might simulate a failure of a critical AWSservice, such as Lambda, API Gateway, or DynamoDB.**
3. Assign roles and responsibilities: Assign roles and responsibilities for each member of your team. This should include who is responsible for executing the disaster recovery plan, who is responsible for communicating with stakeholders, and who is responsible for making decisions related to the incident response.
4. Execute the scenario: Execute the scenario by simulating the disaster and activating your disaster recovery plan. This should be done in a controlled environment to ensure that your production environment is not affected.
5. Review: After the GameDay, review the results with your team. Identify areas for improvement in your disaster recovery plan, as well as any strengths or weaknesses in your team's ability to respond to incidents.

Some specific steps that you might include in a recovery-focused AWS GameDay for serverless infrastructure could include:

- Simulating a failure of a critical AWS service, such as Lambda, API Gateway, or DynamoDB.
- Activating your disaster recovery plan and testing the recovery process for your serverless applications.
- Identifying and addressing any gaps or weaknesses in your recovery plan or processes for serverless infrastructure.
- Reviewing your team's ability to respond to incidents related to serverless infrastructure and identifying areas for improvement.

By conducting a recovery-focused AWS GameDay for serverless infrastructure, you can test and improve your disaster recovery plan, as well as improve your team's ability to respond to incidents and recover from failures related to serverless infrastructure. This can help you ensure the resilience and availability of your serverless applications and services in the event of a disaster or other unexpected event.

AWS FIS

AWS Fault Injection Simulator (FIS) can be used to test the resilience of your serverless applications by injecting faults and failures into your infrastructure. FIS can be used to simulate a variety of failure scenarios, such as network latency, API errors, or AWS service outages.

Here are the basic steps for using AWS FIS with serverless applications:

1. Create an experiment template: First, you need to create an experiment template that defines the fault injection scenario you want to simulate. The template includes details such as the type of failure to inject, the affected resources, and the duration of the experiment.
2. Configure the template for serverless: When creating the experiment template, configure it to target the serverless

resources that you want to test. This could include Lambda functions, API Gateway endpoints, or DynamoDB tables.
3. Create an experiment: Once you have created the experiment template, you can create an experiment instance based on the template. This creates a new experiment that is ready to run.
4. Run the experiment: Start the experiment to inject the fault into your infrastructure. The experiment will run for the specified duration, and you can monitor the results in real-time using AWS CloudWatch metrics.
5. Analyze the results: After the experiment has completed, you can analyze the results to determine how your serverless application responded to the fault injection. You can use the data to identify any weaknesses in your application's resilience and make improvements to your infrastructure.

Some examples of fault injection scenarios you can simulate using AWS FIS with serverless applications include:

- Simulating a Lambda function timeout to test how your application handles function invocations that exceed the configured timeout period.
- Injecting errors into API Gateway responses to test how your application handles unexpected errors from the API.
- Simulating DynamoDB table throttling to test how your application handles throttling and retries when accessing the table.

By using AWS FIS to simulate fault injection scenarios with your serverless applications, you can test and improve the resilience of your applications and infrastructure, helping to ensure that your applications remain available and reliable even in the face of unexpected failures.