



Achieving Serverless Harmony

With Traditional Apps

Mud Ball vs. Spoon

Story Time

- Began with EC2 instances in ASG's, Ansible automation
- Introduced Serverless Framework
- Replace small HTTP service with API Gateway
- Move async jobs into Lambda
- Kill async/cron workers

Expensive Tasks

1. Breaking up shared-memory coupling
2. Connecting legacy apps to new event streams
3. Rewriting jobs in unsupported languages



Incidental Glue

FaaS in other Services

Using Cognito triggers, Config Rules, and infra cron jobs.

Backend Tasks

Intentional Event Stream

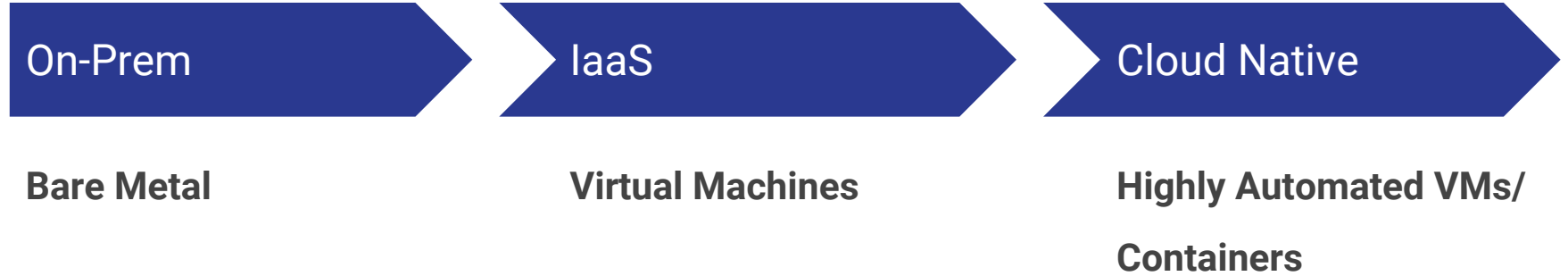
Architecture and plans change to match new tools/processes and intentionally start migrating features.

User-Facing

Major Replacement

Critical-path features move from your monolith to separate serverless deployments.

Starting Points



Hard Parts

Scaling

Downstream Service Considerations

- Introduce back pressure
- Prioritize user-facing tasks
- Denormalize data into messages



Queueueueueueeing

- Kinesis retries handle capacity problems
- Serial, but only if you fit on one shard
- Put the full information the event needs in its trigger
 - S3 notably can't
 - SNS partially does this with invoke event size limit
- Dead Letter Queue

DynamoDB

- Makes a great scratch space because events include
 - HASH+RANGE
 - New and old versions of modified data
 - TTL to expire finished job records while leaving an audit trail
- RCU/WCU remaining provides back pressure
- Adjustable capacity (with limits)
 - Even adjustable ... from a Lambda

Monitoring

Revealing Brewing Problems

- Log log log log
- Cloudwatch connections
- Combine legacy and new sources
- Teach new techniques

Structured Logging

- Logs are events, not strings
- Machine-readability matters
- CloudWatch filters can parse logged values to metrics
- JSON avoids regex pain and sadness

Structured Log

```
{  
  "timestamp": 1493214552,  
  "event": "user.login.success",  
  "user": "ryansb",  
  "method": "password",  
  "source": {  
    "type": "web.navbar",  
    "client": "git-b263b0"  
  }  
}
```

Central Visibility

- Forward CloudWatch Logs to legacy system
- Send application metrics to CloudWatch metrics
- Forward everything to...
 - Splunk
 - ELK stack
 - Graylog

Codebases

- Monorepo vs. split service
- Deployment pipelines
- Testing and staging

Monorepo

- Keeps attribution
- Single history
- Always test full repo
- Cross-cutting changes are easier
- No dependent pull requests
- Smaller individual repos
- Per-repo test pipelines
- Fewer changes in-flight
- Cross-repo changes
- Submodules (ew)

Deploy Pipelines

- Deploys **have** to be independent
- Avoid backwards incompatible changes in event formats
- DB migrations now have one more step

Thanks!

harmony.serverless.zone

@ryan_sb