AARHUS UNIVERSITET

Microservices and DevOps

DevOps and Container Technology

Swarm Deployment

Henrik Bærbak Christensen

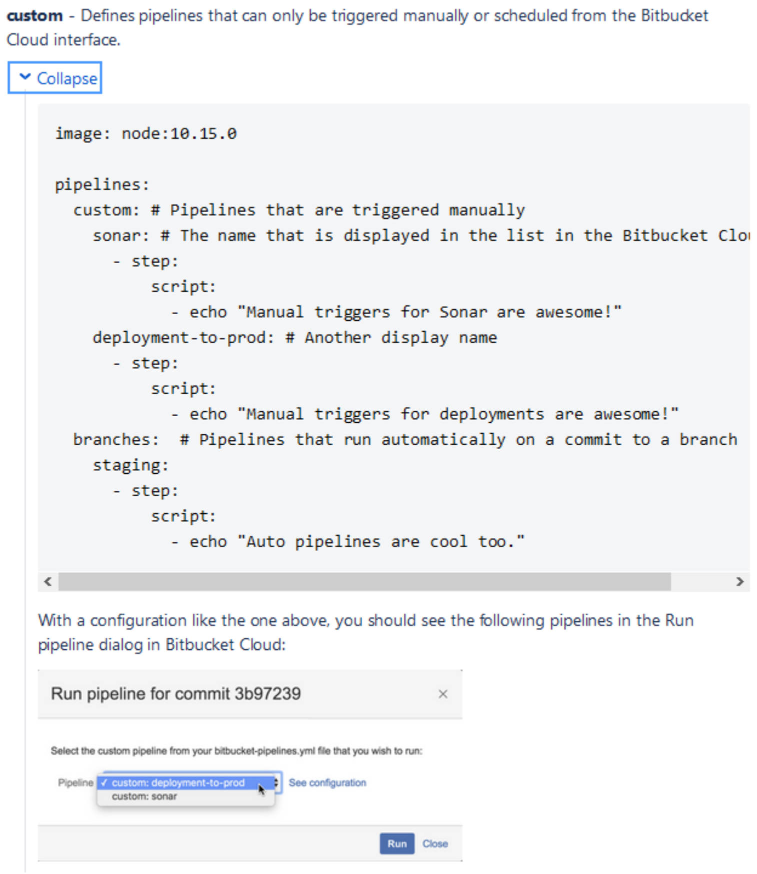
So…

- **Deliver** = **Deploy Release** in **Production**

- When
  - Human gate: When I say ‘go’
  - Human gate: When canaries have survived N hours
  - Automatic: Final stage of the deployment pipeline
    - Push based: Execute commands to deliver
      - E.g. call the ‘deployment interface’ with (version, production, service)
    - Pull based: Just flag version to publish
      - E.g. cron job on production server that monitor flag

BitBucket Pipelines

- Define human gates



custom - Defines pipelines that can only be triggered manually or scheduled from the Bitbucket Cloud interface.

```
image: node:10.15.0

pipelines:
  custom: # Pipelines that are triggered manually
    sonar: # The name that is displayed in the list in the Bitbucket Clo
      - step:
          script:
            - echo "Manual triggers for Sonar are awesome!"
    deployment-to-prod: # Another display name
      - step:
          script:
            - echo "Manual triggers for deployments are awesome!"
  branches:  # Pipelines that run automatically on a commit to a branch
    staging:
      - step:
          script:
            - echo "Auto pipelines are cool too."
```

With a configuration like the one above, you should see the following pipelines in the Run pipeline dialog in Bitbucket Cloud:

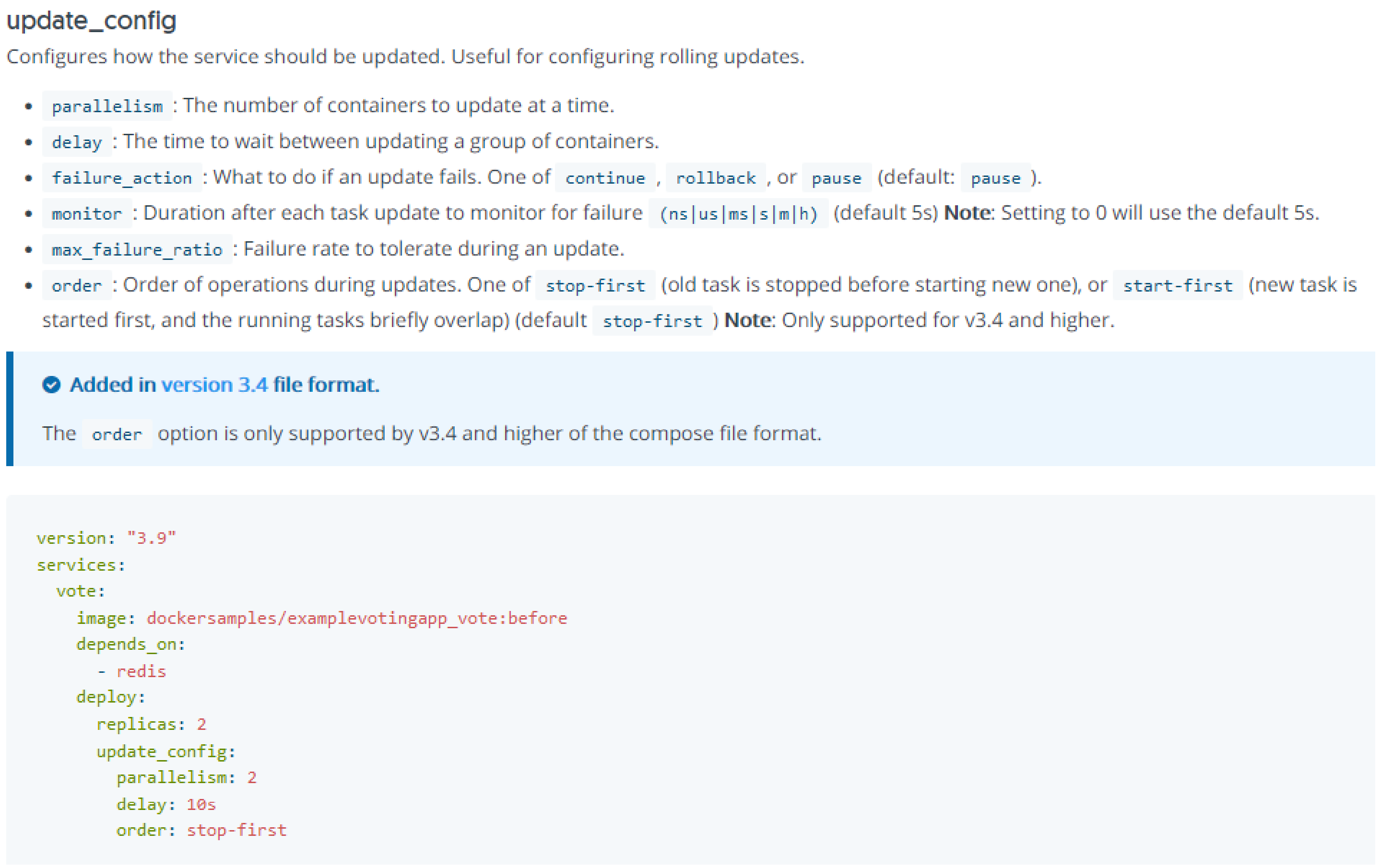
Run pipeline for commit 3b97239

Select the custom pipeline from your bitbucket-pipelines.yml file that you wish to run:

Pipeline [✓ custom: deployment-to-prod] custom: sonar

AARHUS UNIVERSITET

Swarm: update_config



update_config

Configures how the service should be updated. Useful for configuring rolling updates.

- `parallelism` : The number of containers to update at a time.
- `delay` : The time to wait between updating a group of containers.
- `failure_action` : What to do if an update fails. One of `continue` , `rollback` , or `pause` (default: `pause`).
- `monitor` : Duration after each task update to monitor for failure `(ns|us|ms|s|m|h)` (default 5s) **Note**: Setting to 0 will use the default 5s.
- `max_failure_ratio` : Failure rate to tolerate during an update.
- `order` : Order of operations during updates. One of `stop-first` (old task is stopped before starting new one), or `start-first` (new task is started first, and the running tasks briefly overlap) (default `stop-first`) **Note**: Only supported for v3.4 and higher.

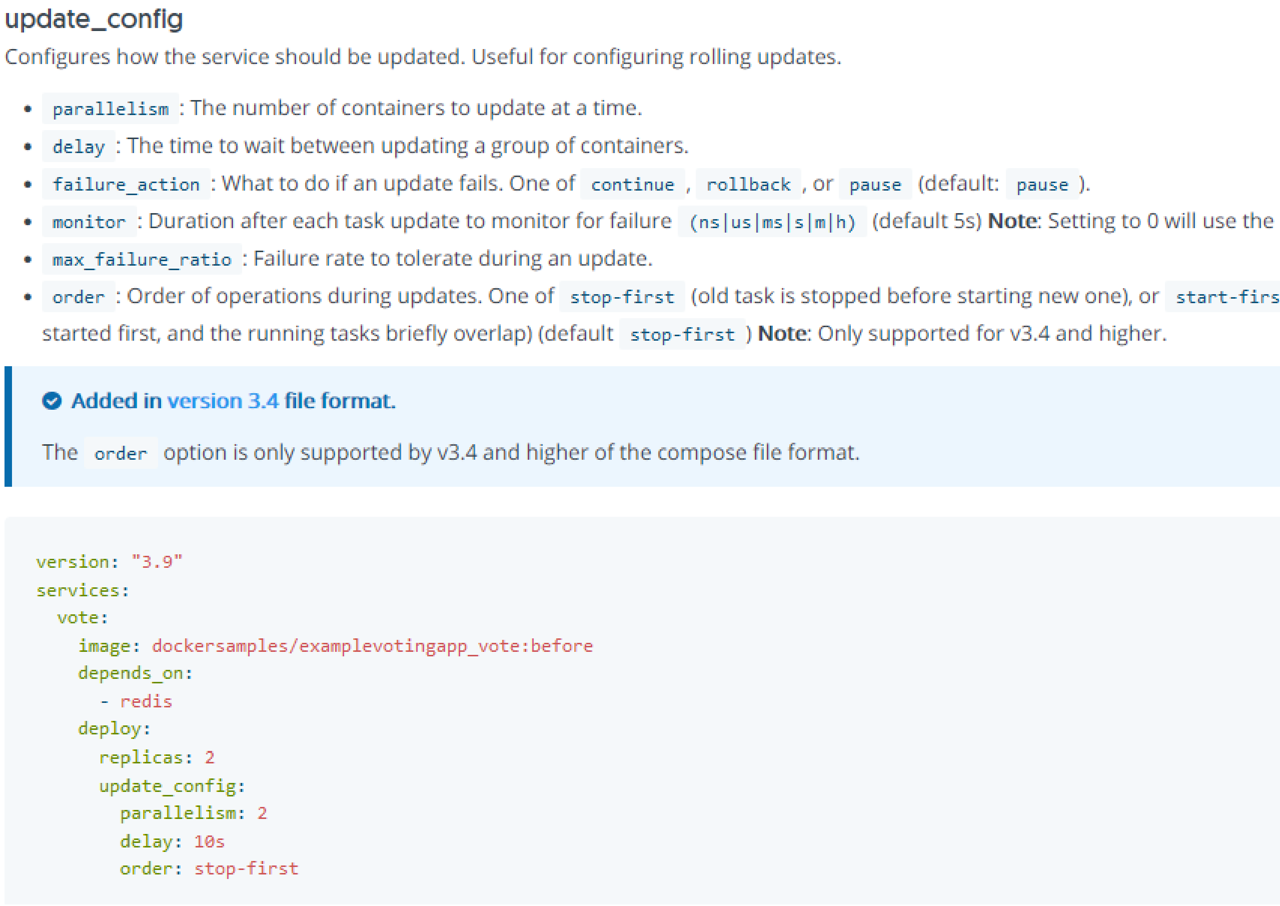
Added in version 3.4 file format.

The `order` option is only supported by v3.4 and higher of the compose file format.

```yaml
version: "3.9"
services:
  vote:
    image: dockersamples/examplevotingapp_vote:before
    depends_on:
      - redis
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
        delay: 10s
        order: stop-first
```

Exercise

- Can we define:
  - Blue/green
  - Canary
  - Rolling



update_config

Configures how the service should be updated. Useful for configuring rolling updates.

- `parallelism` : The number of containers to update at a time.
- `delay` : The time to wait between updating a group of containers.
- `failure_action` : What to do if an update fails. One of `continue` , `rollback` , or `pause` (default: `pause`).
- `monitor` : Duration after each task update to monitor for failure `(ns|us|ms|s|m|h)` (default 5s) **Note**: Setting to 0 will use the
- `max_failure_ratio` : Failure rate to tolerate during an update.
- `order` : Order of operations during updates. One of `stop-first` (old task is stopped before starting new one), or `start-firs` started first, and the running tasks briefly overlap) (default `stop-first`) **Note**: Only supported for v3.4 and higher.

Added in version 3.4 file format.

The `order` option is only supported by v3.4 and higher of the compose file format.

```
version: "3.9"
services:
  vote:
    image: dockersamples/examplevotingapp_vote:before
    depends_on:
      - redis
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
        delay: 10s
        order: stop-first
```

Discussion

- Experience in the audience?

- The Return on Investment
  - Balancing effort on complex ‘deployment interfaces’ development versus roll out schedule and monitoring